UNIVERSITÉ DU QUÉBEC À MONTRÉAL

DÉTECTION DES TRANSFERTS HORIZONTAUX DE GÈNES : MODÈLES ET ALGORITHMES APPLIQUÉS À L'ÉVOLUTION DES ESPÈCES ET DES LANGUES

THÈSE
PRÉSENTÉE
COMME EXIGENCE PARTIELLE
DU DOCTORAT EN INFORMATIQUE

PAR

ALIX BOC

JANVIER 2012

REMERCIEMENTS

Je tiens à exprimer mes plus vifs remerciements au Dr. Vladimir Makarenkov, mon directeur de recherche, pour sa patience sans égale, ses conseils, suggestions et ses contributions pour la réalisation de ce projet de recherche. Qu'il trouve ici toute l'expression de ma gratitude et de ma profonde reconnaissance.

Je tiens également à remercier mes collègues du laboratoire de bioinformatique : tous ensemble nous avons formé un groupe dynamique que ce soit pour la recherche ou pour des activités en dehors de l'université. Cette ambiance de travail a été très bénéfique.

À tous ceux qui ont contribué de près ou de loin à la réalisation de ce travail, qu'ils trouvent ici mes plus vifs remerciements.

Je désirerais aussi remercier les professeurs qui ont lu cette thèse. Je leur suis reconnaissant du temps qu'ils ont pris pour lire cette étude et pour les commentaires qu'ils ont apportés.

Sur un plan plus personnel, je ne peux oublier de remercier mon épouse Joanne, pour sa patience et son soutien moral, ma mère, mes frères et ma sœur, les autres membres de ma famille, ainsi que tous mes amis qui m'ont soutenu tout au long de mon parcours. Une pensée spéciale pour mon père qui nous a quitté depuis longtemps et qui est toujours très présent dans nos mémoires.

À mes deux nièces Christina Roza et Cynthia Roza

TABLE DES MATIÈRES

LISTE DES FIGURES xi
RÉSUMÉ xv
INTRODUCTION 1
CHAPITRE I
NOTIONS DE BASE 3
1.1 Introduction 3
1.2 Un arbre phylogénétique 3
1.3 La reconstruction d'arbres phylogénétiques 5
1.4 L'évolution réticulée 7
1.5 Le transfert horizontal de gènes 8
1.5.1 La conjugaison 9
1.5.2 La transformation 10
1.5.3 La transduction 10
1.5.4 Le transfert horizontal de gènes chez les eucaryotes 10
1.5.5 Impact d'un THG sur une phylogénie 11
CHAPITRE II
DÉTECTION DES TRANSFERTS HORIZONTAUX DE GÈNES : ÉTAT DE L'ART 13
2.1 Introduction 13
2.2 L'évolution réticulée 13
2.3 Les méthodes de détection de transferts horizontaux de gènes 14
2.3.1 L'algorithme *LatTrans* 15
2.3.2 L'algorithme *HGT-Detection* 16
2.3.3 Autres travaux dans le domaine des transferts horizontaux 23
2.4 Les objectifs du projet doctoral 24

CHAPITRE III
ALGORITHME POUR LA DÉTECTION DES TRANSFERTS HORIZONTAUX DE GÈNES COMPLETS 25
3.1 Introduction 25
3.2 Nouvel algorithme pour l'inférence et la validation des THG 26
3.2.1 La dissimilarité de bipartitions et autres critères d'optimisation 28
3.2.2 La contrainte de sous-arbres 31
3.2.3 Algorithme d'inférence des transferts horizontaux de gènes complets 37
3.2.4 Validation des transferts horizontaux de gènes 39
3.3 Simulations Monte-Carlo 42
3.3.1 Description des simulations 42
3.3.2 Résultats des simulations 45
3.4 Exemples 56
3.4.1 Détection des transferts horizontaux du gène *rpl12e* 56
3.4.2 Détection des transferts horizontaux de *PheRS* synthétase 62
3.5 Discussion et conclusion 66
CHAPITRE IV
MODÈLE DU TRANSFERT HORIZONTAL PARTIEL 73
4.1 Introduction 73
4.2 Les gènes mosaïques 74
4.3 Premier modèle d'inférence des transferts partiels 76
4.4 Deuxième modèle d'inférence des transferts partiels 83
4.4.1 Algorithme pour détecter des transferts partiels 84
4.4.2 Simulations Monté-Carlo 87
4.4.3 Exemples 90
4.5 Discussion et conclusion 99
CHAPITRE V
APPLICATION DE L'ALGORITHME DE DÉTECTION DES TRANSFERTS HORIZONTAUX À L'ETUDE DE L'ÉVOLUTION DES LANGUES INDO-EUROPÉENNES 103
5.1 Introduction 103

5.2 Application de l'algorithme de détection des THG pour modéliser les emprunts de mots entre les langues Indo-Européennes 104
5.2.1 Description des données 104
5.2.2 Méthodologie 105
5.3 Résultats et discussion 110
CONCLUSION ET PERSPECTIVES 117
ANNEXE A
EXEMPLES D'INTERFACES WEB DU LOGICIEL *T-REX* 123
ANNEXE B
DOCUMENTS SUPPLÉMENTAIRES RELATIFS AU CHAPITRE III 131
B.1 Schéma algorithmique de *HGT-Detection* (Boc *et al.*, 2010) 131
B.2 Trace d'exécution de l'algorithme *RIATA-HGT* appliqué au jeu de données du gène *rpl12e* 132
B.3 Trace d'exécution de l'algorithme *RIATA-HGT* appliqué au jeu de données du gène *PheRS synthétase* 132
ANNEXE C
EXEMPLES DE CODE SOURCE 135
C.1 Script PERL pour la détection des transferts partiels 135
C.2 Script PERL pour la détection des transferts complets 143
C.3 Programme principal pour la détection des transferts complets 151
C.4 Fonction permettant la détection de plusieurs transferts indépendants par itérations .163
ANNEXE D
ARTICLES PUBLIES POUR PUBLICATION DANS LE CADRE DU PROJET DOCTORAL 167
GLOSSAIRE 285
BIBLIOGRAPHIE 289

LISTE DES FIGURES

Figure Page

1.1 Modèle de base introduisant un arbre phylogénétique....4

1.2 Exemple d'une distance d'arbre sur un ensemble X de 5 taxons et l'arbre phylogénétique associé. 6

1.3 Un réseau réticulé....7

1.4 Le réseau réticulé représenterait mieux l'histoire de la vie qu'un arbre phylogénétique classique (Doolittle, 1999)....8

1.5 Trois mécanismes de transfert horizontal de gènes. 9

1.6 Incongruence des arbres de gène par rapport à l'arbre d'espèce, tiré de Philippe *et al.* (2003)....12

2.1 Scénario de transferts du gène *rbcL* identifié par Hallet et Lagergren (2001)....16

2.2 Comparaison des deux modèles de transferts horizontaux. 18

2.3 Modèle d'évolution impliquant des transferts partiels....19

2.4 La distance de Robinson et Foulds entre T et T_1 est égale à 2. 21

2.5 Un résultat du programme *HGT-Detection* inclus dans la version Web de T-Rex....22

3.1 Cas de figures où un transfert de gène est interdit. 27

3.2 Les arbres T et T' et leur table de bipartitions....29

3.3 Illustration de la contrainte de sous-arbres. 32

3.4 Le transfert entre les arêtes (x,y) et (z,y) fait partie du scénario de coût minimal....33

3.5 Le transferts entre les arêtes (x,y) et (z,w) fait partie du scénario de coût minimal transformant T en T' si toutes les bipartitions des arêtes du chemin (x',z') dans l'arbre d'espèces transformé T_1 sont présentes dans la table de bipartitions de T' et que le sous-arbre Sub_{yw} est présent dans T'. 36

3.6 Taux de détection des transferts horizontaux en fonction du nombre de THG pour quatre niveaux de confiance des arbres de gène. 46

3.7 Comparaison des stratégies algorithmiques utilisant différents critères d'optimisation (RF, LS, BD et QD). 48

3.8 Comparaison de la stratégie basée sur BD (▣) avec *LatTrans* (□). 51

3.9 Comparaison de *HGT-Detection* (□) avec *LatTrans* (□) en termes de temps d'exécution. 52

3.10 Comparaison de *LatTrans* (□) avec *HGT-Detection* (▣) en termes de taux de détection de transferts et de temps d'exécution. 54

3.11 Calcul de la valeur de support de bootstrap d'un transfert horizontal par *RIATA-HGT*. 56

3.12 Arbre de maximum de vraisemblance du gène *rpl12e*. 57

3.13 Scénario de transferts obtenu par l'algorithme *HGT-Detection* appliqué au jeu de données du gène *rpl12e* 59

3.14 Scénarios de transferts obtenus par l'algorithme *RIATA-HGT* appliqué au jeu de données du gène *rpl12e* 61

3.15 Arbre phylogénétique du *PheRS* inféré avec *PHYML* 63

3.16 Scénario de transferts obtenu par l'algorithme *HGT-Detection* appliqué aux séquences du *PheRS* synthétase. 65

3.17 Scénarios de transferts obtenus par l'algorithme *RIATA-HGT* appliqué aux séquences du *PheRs* synthétase. 68

3.18 Une opération SPR transformant l'arbre d'espèces T en l'arbre de gène T'. 69

4.1 Un gène mosaïque incluant une sous-séquence (en blanc) provenant d'une autre espèce. 75

4.2 La situation où le transfert (b,a) affecte la distance $d(i,j)$, mais pas la distance $d(i_1,j)$. . 76

4.3 Situations où la distance évolutive entre i et j ne change pas après l'ajout de la nouvelle arête (b,a). 77

4.4 Les transferts croisés doivent être interdits. 79

4.5 La distance i,j peut être affectée par les deux transferts uniquement dans les cas (a) et (b). 80

4.6 L'arbre de gène partiel est inféré en utilisant les séquences situées à l'intérieur de la fenêtre coulissante et l'algorithme de détection des transferts complets, incluant l'étape de validation par bootstrap, est alors appliqué. 86

4.7 Taux de détection et taux de faux positifs en fonction du nombre de feuilles et du nombre de transferts horizontaux partiels. 89

4.8 Taux moyens de détection (vrais et faux positifs) pour les cas de 1 à 5 transferts partiels générés. 90

4.9 La phylogénie du gène *rbcL* pour 42 bactéries et plastides obtenue à l'aide de *PHYML*. 91

4.10 Les transferts complets obtenus en appliquant l'algorithme *HGT-Detection*. 93

4.11 Les transferts partiels du gène *rbcL* obtenus en appliquant le deuxième algorithme de détection de THG partiels. 95

4.12 La phylogénie du gène *mutU* (Denamur *et al.*, 2000). 97

4.13 Les transferts partiels du gène *mutU* retrouvés par Denamur *et al.* (2000). 97

4.14 Les transferts partiels du gène *mutU* trouvés par le deuxième algorithme de détection des THG partiels. Les score de bootstrap et les intervalles de l'ASM affectés sont indiqués à côté de chaque transfert. 99

5.1 L'arbre d'évolution des langues IE pour 14 groupes principaux. 106

5.2 La distance topologique de Robinson et Foulds normalisée entre chaque arbre de mot (1484 au total) et l'arbre de langues réduit correspondant. 107

5.3 Deux exemples d'emprunts impliquant plusieurs groupes de langues. 109

5.4 Évolution présumée du mot FRUIT. 112

5.5 Résultats obtenus pour les mots des catégories lexicale et fonctionnelle (i.e., le total des mots). 113

5.6 Résultats obtenus pour les mots de la catégorie lexicale. 114

5.7 Résultats obtenus pour les mots de la catégorie fonctionnelle. 115

A.1 Interface principale de la version Web de *T-Rex*. 123

A.2 Interface de *HGT-Detection*. 124

A.3 Page des résultats de *HGT-Detection*. 125

A.4 Exemple de solution à 5 transferts horizontaux, détectés et affichés avec le tracé hiérachique horizontal de l'arbre. 126

A.5 Exemple de solution à 5 transferts horizontaux, détectés et affichés avec le tracé hiérachique vertical de l'arbre. 127

A.6 Exemple de solution à 5 transferts horizontaux, détectés et affichés avec le tracé axial de l'arbre. 128

A.7 Exemple de solution à 5 transferts horizontaux, détectés et affichés avec le tracé radial de l'arbre. 129

A.8 Exemple de solution à 9 transferts horizontaux, détectés et affichés avec le support de bootstrap en utilisant un tracé hiérachique horizontal de l'arbre. 130

RÉSUMÉ

Le transfert horizontal de gènes (THG, ou transfert latéral de gènes) est un mécanisme d'évolution naturel qui consiste en le transfert direct du matériel génétique d'une espèce à une autre. La possibilité que le transfert horizontal de gènes puisse jouer un rôle clé dans l'évolution biologique est un changement fondamental dans notre perception des aspects généraux de la biologie évolutive survenu ces dernières années. Par exemple, les bactéries et les virus possèdent des mécanismes sophistiqués d'acquisition de nouveaux gènes par transfert horizontal leur permettant de s'adapter et d'évoluer adéquatement dans leur environnement. Jusqu'à tout récemment, les méthodes de détection de ce mécanisme reposaient essentiellement sur l'analyse de séquences et étaient très rarement automatisées. Il est impossible de représenter l'évolution d'organismes ayant subi des THG à l'aide d'arbres phylogénétiques acycliques. La présentation adéquate est celle d'un réseau. Dans cette thèse, nous décrivons un nouveau modèle de ce mécanisme d'évolution, en se basant sur l'étude de différences topologiques et métriques entre un arbre d'espèces et un arbre du gène inférés pour le même ensemble d'espèces. Les méthodes qui en découlent ont été appliquées à des jeux de données réelles où des hypothèses de transferts latéraux de gènes étaient plausibles. Des simulations Monté-Carlo ont été menées afin d'évaluer la qualité des résultats par rapport à des méthodes existantes. Nous présentons également une généralisation du modèle de transferts horizontaux complets qui est applicable pour détecter des transferts partiels et identifier des gènes mosaïques. Dans ce dernier modèle, on suppose qu'une partie seulement du gène a été transférée. Enfin, nous présentons une application de ces nouvelles méthodes servant à modéliser des emprunts de mots survenus durant l'évolution des langues indo-européennes.

Mots clés : arbre phylogénétique, réseau réticulé, transfert horizontal de gènes, critère des moindres carrés, distance de Robinson et Foulds, dissimilarité de bipartitions, biolinguistique.

INTRODUCTION

Cette thèse porte sur la conception de nouveaux algorithmes permettant la détection de transferts horizontaux d'un gène pour un groupe d'espèces étudiées. Le transfert horizontal de gènes (THG), appelé aussi transfert latéral de gènes (TLG), est un mécanisme naturel qui permet à des organismes, notamment à des bactéries et à des virus, de s'échanger des gènes. C'est un processus dans lequel un organisme intègre le matériel génétique provenant d'un autre organisme qui n'est pas son descendant direct. Par opposition, le transfert vertical se produit lorsque l'organisme reçoit du matériel génétique à partir de son ancêtre le plus proche. Les différentes techniques existantes de détection des THG sont basées sur l'analyse de la composition nucléotidique des séquences biologiques et sur des conflits entre phylogénies permettant aux biologistes d'émettre une hypothèse de transfert horizontal de gène. Ces procédés n'étaient pas toujours automatisés et ne comprenaient pas une étape de validation des transferts obtenus.

Dans cette thèse, nous présentons une approche algorithmique à ce problème qui utilise le principe de réconciliation d'arbres d'espèces et de gène*, tout en considérant des contraintes d'évolution biologiques. Nous aurons recours à des algorithmes heuristiques pour maintenir la complexité algorithmique polynomiale, tout en préservant la viabilité des résultats.

Pour mieux comprendre la problématique de la détection de transferts horizontaux, nous présentons dans le chapitre I, les notions de bases nécessaires à la compréhension de ce mécanisme évolutionnaire. Nous décrivons alors, brièvement, la phylogénie (i.e., arbre phylogénétique), les méthodes de reconstruction d'arbres phylogénétiques, les arbres réticulés et enfin, le transfert horizontal de gènes.

---

* Le singulier est utilisé dans le terme « arbre de gène » car il s'agit de la phylogénie construite à la base d'un seul gène.

Dans le chapitre II, nous ferons un état de l'art des méthodes et des logiciels existants pour la reconstruction et la visualisation d'arbres réticulés et plus particulièrement pour la détection de transferts horizontaux de gènes.

Après cette revue de la littérature, nous présentons au chapitre III notre contribution principale. La méthode que nous proposons traite de la question du transfert complet qui suggère que la séquence transférée entre deux espèces représente un gène au complet. Nous y décrivons un nouvel algorithme, une nouvelle mesure de comparaison d'arbres phylogénétiques et un procédé de validation des résultats. Des simulations Monte-Carlo et des tests sur des jeux de données réels menés pour évaluer l'efficacité du nouvel algorithme seront également présentés. Ce projet de recherche a mené à deux publications dans les actes des conférences (Makarenkov *et al.*, 2006 et Makarenkov *et al.*, 2007) et d'une publication plus globale dans la revue *Systematic Biology* (Boc *et al.*, 2010a).

Le chapitre IV décrit une généralisation de notre solution. En effet, nous y traitons le problème du transfert partiel qui suppose que seule une partie d'un gène donné peut être transférée. Bien que cette seconde approche semble être une généralisation de la première, des considérations de complexité algorithmique motivent la nécessité de traiter les deux cas séparément. Ce projet de recherche a mené à trois publications (Makarenkov *et al.*, 2006, Makarenkov *et al.*, 2008 et Boc *et al.*, 2011, soumis à *Nucleic Acids Research*).

Dans le chapitre V, nous présentons une application de nos méthodes de détection de transferts dans un domaine connexe. Précisément, nous les appliquons à l'étude de l'évolution des langues indo-européennes. L'évolution des langues est souvent représentée à l'aide des arbres phylogénétiques, mais les linguistes omettent souvent les mots empruntés d'une langue à l'autre pour éviter les interférences dans la représentation des résultats. Dans le modèle d'évolution des langues naturelles que nous proposons, nous prenons en compte des emprunts linguistiques survenus au cours de l'évolution. Ce projet de recherche a fait l'objet d'une publication dans les actes de la conférence IFCS-2009 (Boc *et al.*, 2010b).

En conclusion, nous faisons une synthèse du travail effectué et présentons quelques pistes possibles pour l'amélioration des résultats obtenus.

CHAPITRE I

NOTIONS DE BASE

1.1 Introduction

Afin de proposer une solution originale et efficace au problème de la détection de transferts horizontaux de gène pour un groupe d'espèces observées, nous nous sommes intéressés à différents sujets d'études que nous présentons brièvement dans ce chapitre. Tout d'abord nous verrons les éléments de bases de la reconstruction d'arbres phylogénétiques, soient les approches et méthodes les plus populaires. Par la suite, nous introduirons la notion d'évolution réticulée, et enfin nous décrirons, d'un point de vue biologique, le processus de transfert horizontal et les différents mécanismes qui le caractérise.

1.2 Un arbre phylogénétique

La phylogenèse étudie la reconstruction de l'histoire évolutive des êtres vivants. Le terme phylogenèse (du grec *phulon*, signifiant "race, tribu") a été introduit par Haeckel en 1860, qui l'a défini comme "*l'histoire du développement paléontologique des organismes par analogie avec l'ontogénie ou histoire du développement individuel*". Un arbre est dit *arbre phylogénétique, phylogénie ou X-arbre* (Barthélémy et Guénoche, 1991) si, dans l'analyse des caractères sur laquelle il repose, le concept de "descendance des espèces avec modification de leurs caractères" a été utilisé. Ce dernier concept signifie que les caractères sont transmis d'une génération à l'autre à travers les mécanismes de l'hérédité impliquant leurs éventuelles modifications (par exemple les mutations). Un arbre phylogénétique est une représentation graphique de la phylogenèse d'un groupe d'espèces (ou de taxons).

Un arbre phylogénétique est composé de quatre principaux éléments. Les feuilles ou nœuds externes représentent les espèces pour lesquelles on dispose de données d'évolution. Les branches (ou arêtes) définissent les relations entre les taxons en termes de descendance. Les nœuds internes sont associés à des ancêtres virtuels. Et enfin, la racine représente l'ancêtre commun de toutes les espèces considérées.

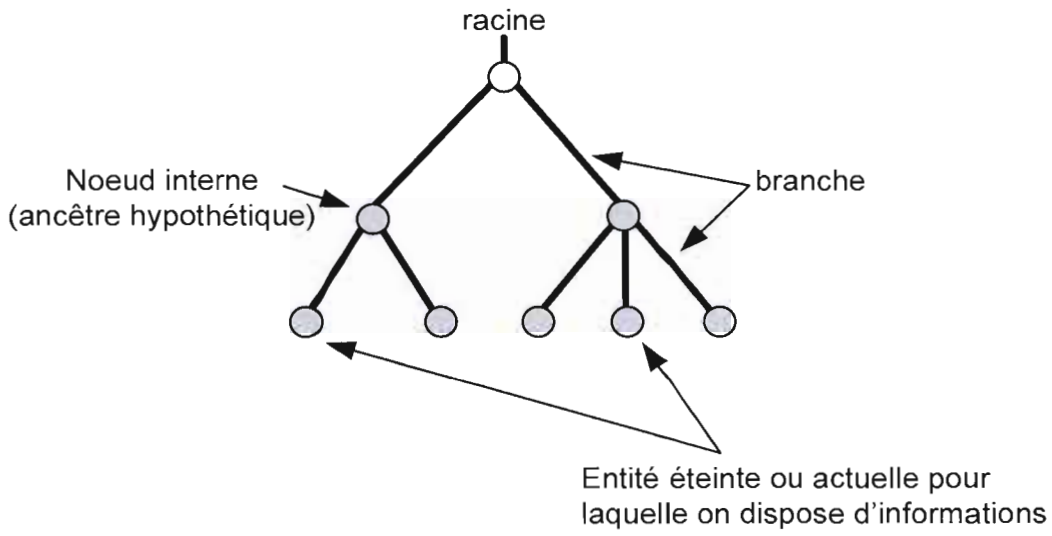


Figure 1.1 Modèle de base introduisant un arbre phylogénétique.

Le degré d'un nœud est le nombre d'arêtes adjacentes à ce nœud. Si ce degré est supérieur à 3, ce nœud est dit non résolu (signifiant la divergence simultanée ou l'incertitude).

On distingue deux types d'arbres. Les arbres enracinés et les arbres non enracinés. Un arbre enraciné est orienté et cette orientation correspond au sens de l'évolution. Il permet donc de définir une relation ancêtre - descendant entre deux nœuds successifs. Dans un arbre non-enraciné, la notion de temps n'existe pas et on ne peut plus définir la relation ancêtre - descendant au niveau des nœuds internes. Ce type d'arbres peut être utilisé lorsque l'on s'intéresse à la classification d'un groupe d'espèces sans considérer le sens d'évolution.

1.3 La reconstruction d'arbres phylogénétiques

La reconstruction d'un arbre phylogénétique commence par l'analyse des séquences nucléotidiques ou d'acides aminés associées aux espèces étudiées. Une séquence nucléotidique (assemblage linéaire de nucléotides) représente l'ADN (acide désoxyribonucléique) et est composée de quatre types de base. Les cytosines (C) et thymines (T) qui font partie de la famille des pyrimidines et les adénines (A) et guanines (G) qui font partie de la famille des purines. Une séquence d'ADN peut représenter un gène qui sera exprimé en une protéine (séquence d'acides aminés).

Il existe trois grandes approches pour construire des arbres phylogénétiques : la cladistique, la phénétique et la probabiliste.

- La cladistique cherche à établir des relations de parenté en s'intéressant aux caractères (bases) dérivés partagés par les taxons ; les méthodes utilisées sont basées sur le maximum de parcimonie.
- La phénétique étudie la parenté entre les taxons en s'intéressant à leur degré de similarité ; les méthodes utilisées sont basées sur les distances.
- La probabiliste ou maximum de vraisemblance évalue, en termes de probabilités, l'ordre des branchements et la longueur des arêtes d'un arbre sous un modèle évolutif donné. Les méthodes bayésiennes font aussi partie de cette approche.

Nous donnons ici quelques définitions de base concernant des arbres phylogénétiques et des métriques d'arbres, en suivant la terminologie de Barthélemy et Guénoche (1988, 1991). La distance $\delta(x,y)$ entre deux sommets x et y dans un arbre phylogénétique (i.e., arbre additif) T est définie comme la somme de toutes les longueurs des arêtes du chemin unique liant x et y dans T. Un tel chemin est noté (x,y). Une feuille est un sommet de degré un.

Définition 1

Soit X un ensemble fini de n taxons. Une *dissimilarité* d sur X est une fonction non-négative sur $(X \times X)$ telle que pour tout x,y appartenant à X :

(1) $d(x,y) = d(y,x)$, et

(2) $d(x,y) = d(y,x) \geq d(x,x) = 0$.

Définition 2

Une dissimilarité d sur X satisfait la condition des quatre points si pour tout x, y, z, et w de X :

$$d(x,y) + d(z,w) \leq \text{Max } \{ d(x,z) + d(y,w); d(x,w) + d(y,z) \}.$$

Définition 3

Pour un ensemble fini X, un arbre phylogénétique (i.e., un arbre additif ou un X-arbre) est une paire ordonnée (T, φ) consistant en un arbre T, avec un ensemble de sommets V et une relation $\varphi: X \rightarrow V$, ayant la propriété que, pour tout $x \in X$ avec un degré d'au moins deux, $x \in \varphi(X)$. Un arbre phylogénétique est binaire si φ est une bijection de X dans l'ensemble de feuilles de T et que chaque sommet interne a un degré égal à 3.

Le théorème principal relatant la condition des quatre points et la représentabilité d'une dissimilarité par un arbre phylogénétique (i.e., une phylogénie) est comme suit :

Théorème 1 (*Zarestskii, Buneman, Patrinos et Hakimi, Dobson*)

Toute dissimilarité satisfaisant la condition des quatres points peut être représentée par un arbre phylogénétique tel que pour tout x,y appartenant à X, $d(x,y)$ est égale à la longueur du chemin liant les feuilles x et y dans T. Cette dissimilarité est appelée une distance d'arbre. Cet arbre est unique.

	x2	x3	x4	x5
x1	6	6	4	2
x2		2	4	6
x3			4	6
x4				4

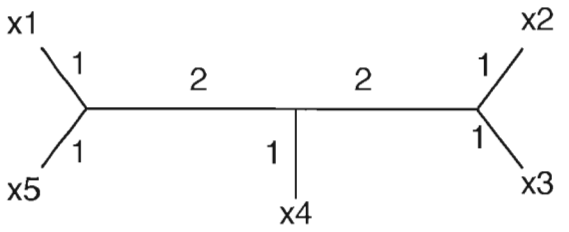


Figure 1.2 Exemple d'une distance d'arbre sur un ensemble X de 5 taxons et l'arbre phylogénétique associé.

1.4 L'évolution réticulée

Plusieurs importants mécanismes phylogénétiques s'expliquent par le phénomène de l'évolution réticulée qui suppose des liens supplémentaires entre les espèces par rapport au modèle arborescent classique (Doolittle, 1999, Legendre 2000). L'évolution réticulée reflète la part de l'évolution des espèces qui ne peut pas être représentée correctement par le modèle de bifurcation utilisé classiquement en analyse phylogénétique. La figure 1.3 montre un arbre réticulé (ici un réticulogramme). Le trait ajouté entre les arêtes 1 et 2 représente une arête de réticulation ajoutée à l'arbre original.

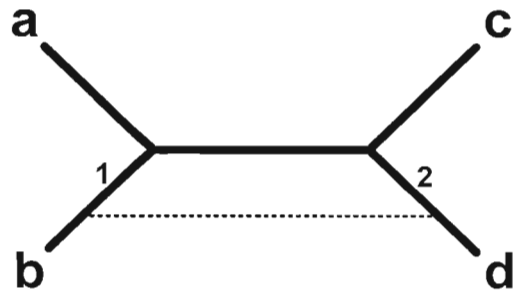


Figure 1.3 Un arbre réticulé.

Dans son célèbre article, Doolittle (1999) a mis l'accent sur le rôle de l'évolution réticulée, et plus précisément du transfert horizontal des gènes, dans l'évolution des bactéries, de même que des espèces plus complexes. La figure 1.4 présentée par Doolittle montre que l'évolution des espèces se produit selon un modèle en réseau plutôt qu'un modèle en arbre. D'autre part, les spécialistes en biologie évolutive ont remarqué que des phénomènes très importants, tels que l'hybridation et l'allopolyploïdie, ne correspondent pas au modèle d'évolution arborescente (Legendre, 2000, Lapointe, 2000). Le modèle d'évolution en réseau a été utilisé dans tous nos algorithmes pour la détection de transferts horizontaux de gènes décrits dans cette thèse.

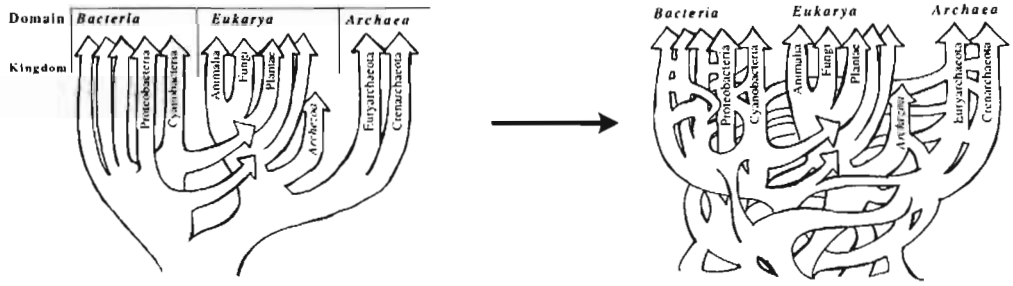


Figure 1.4 Le réseau réticulé représenterait mieux l'histoire de la vie qu'un arbre phylogénétique classique (Doolittle, 1999).

1.5 Le transfert horizontal de gènes

Le transfert horizontal de gènes (THG) est un des mécanismes majeurs contribuant à la diversification des génomes microbiens. Le THG est dominant parmi les groupes variés de procaryotes (Doolittle, 1999). La compréhension du rôle clé joué par le THG dans l'évolution des espèces a été l'un des changements les plus fondamentaux dans notre perception de l'aspect général de la biologie évolutive (Doolittle *et al.*, 2003; Koonin, 2003). Le THG peut poser plusieurs risques pour l'humain, incluant : l'insertion d'ADN transgénique dans les cellules humaines qui déclenche le cancer, des gènes résistants aux antibiotiques qui se propagent parmi des bactéries pathogènes, des gènes associés à des maladies se propageant et se recombinant pour créer de nouveaux virus ou de nouvelles bactéries (Ho, 2002). Il existe trois principaux mécanismes, illustrés sur la figure 1.5, de transfert horizontal de gènes chez les bactéries : la conjugaison, la transformation et la transduction (Bauman, 2005).

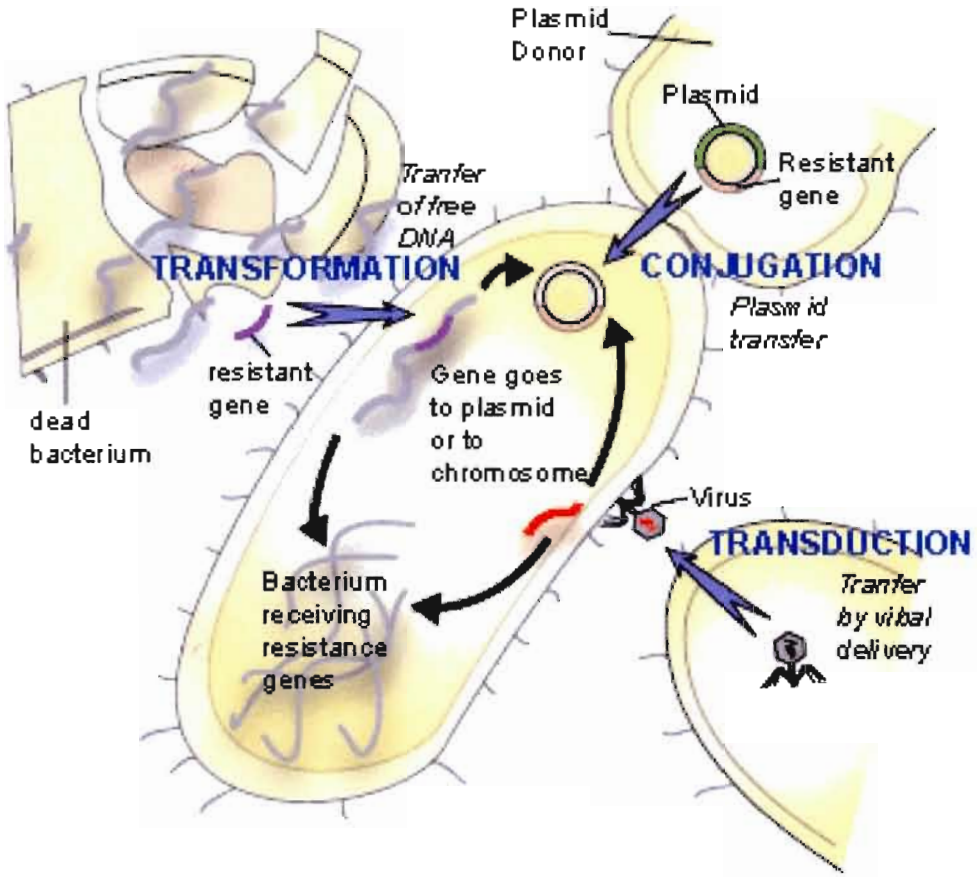


Figure 1.5 Trois mécanismes de transfert horizontal de gènes incluant la transformation, la conjugaison et la transduction.

1.5.1 La conjugaison

Le premier mécanisme est celui de la conjugaison où les organismes mettent au point un système qui leur permet de s'échanger du matériel génétique pour s'adapter à leur environnement. On peut observer ce phénomène lorsqu'une bactérie entre en contact avec un antibiotique hostile. Certains gènes subissent des mutations pour résister à l'antibiotique. Ces gènes sont par la suite transmis à d'autres bactéries. Le processus est le suivant : deux cellules entrent en contact et s'échangent une partie de leur matériel génétique via le cytoplasme. Ce mécanisme est particulièrement important entre des organismes d'une même espèce, mais il peut également avoir lieu entre des organismes faisant partie d'espèces différentes.

1.5.2 La transformation

La transformation est le mécanisme le plus simple. Dans le milieu extérieur d'un organisme se trouvent des fragments d'ADN libre qui résultent en général de la mort d'un autre organisme. Un tel fragment d'ADN libre peut être intégré à l'intérieur d'une cellule particulière de l'hôte, et puis intégré au génome entier de l'hôte. Il y a donc un transfert horizontal entre deux espèces qui peuvent être différentes.

1.5.3 La transduction

La transduction est un transfert de matériel génétique (ADN chromosomique ou extra-chromosomique ou ARN) par des bactériophages (ou virus attaquant les bactéries), dits transducteurs. Compte tenu de l'étroite spécificité existant entre les phages et les bactéries, ces transferts se font essentiellement entre des bactéries appartenant à une même espèce ou à des espèces apparentées. Grâce aux phages, la transduction peut cependant se produire entre des souches bactériennes appartenant à des espèces phylogénétiquement éloignées.

Ces trois mécanismes sont très fréquents chez les procaryotes et génèrent souvent des échanges massifs de matériels génétiques (Jain *et al.*, 1999; Lake et Rivera, 2007). Néanmoins, le matériel génétique n'est pas forcément conservé par l'organisme hôte. Le gène ou le complexe de gènes transférés horizontalement doit être avantageux pour l'hôte. Dans la plupart des cas, suite à un transfert horizontal de matériel génétique d'une espèce à l'autre, l'ADN correspondant ne procure pas d'avantage sélectif et le nouveau gène est rejeté par l'hôte. Dans d'autres cas plutôt rares, il y a l'acquisition, grâce à un gène transféré horizontalement, d'une nouvelle fonction (Keeling, 2009 ; Lawrence, 1999) ou d'une résistance aux antibiotiques. Dans le dernier cas, le gène transféré est conservé dans la population de l'hôte.

1.5.4 Le transfert horizontal de gènes chez les eucaryotes

Il existe aussi deux types distincts de transfert de gènes chez les eucaryotes : le transfert de gènes à partir des organites d'origine endosymbiotique dans le noyau de la cellule eucaryote (transfert de gènes endosymbiotique) et le transfert de gènes entre espèeces non-apparentées (Anderson, 2005).

Le transfert de gènes endosymbiotiques est largement reconnu comme une source importante de matériel génétique dans des lignées d'eucaryotes (Timmis *et al.*, 2004). Tout d'abord, le matériel génétique étranger doit entrer dans la cellule, soit comme de l'ADN nu, soit avec la cellule qui abrite le gène. Une fois à l'intérieur, le gène doit être incorporé dans le noyau hôte et puis exprimer une protéine fonctionnelle. Pour que le matériel génétique étranger soit maintenu, la protéine doit produire une fonction qui est sélectionnée dans la population affectée. Intuitivement, il peut sembler très peu probable qu'un gène procaryote puisse être transféré avec succès à un eucaryote. Cependant, l'incorporation de matériel génétique étranger dans le noyau des eucaryotes se produit à un taux important ; une grande quantité de matériel génétique des mitochondries a été incorporée dans les génomes d'eucaryotes, dont une partie a été montré fonctionnelle (Adams et Palmer, 2003).

La présence de THG chez les eucaryotes est beaucoup plus controversée que chez les procaryotes. Doolittle (1998) a présenté l'hypothèse "vous êtes ce que vous mangez" pour décrire le mécanisme d'échanges par lequel les eucaryotes phagotrophes se nourrissant de bactéries pourraient intégrer des parties de leur matériel génétique au cours de l'évolution. Des cas de transferts de gènes ont été observés chez des eucaryotes non-phagotrophes, ce qui impliquerait d'autres mécanismes tels que le transfert par virus ou le contact physique impliquant des relations symbiotiques ou hôtes-parasites (Gogarten, 2003).

1.5.5 Impact d'un THG sur une phylogénie

De nombreuses incongruences peuvent être observées entre des phylogénies de gène et d'espèces. La figure 1.6 présente une phylogénie d'espèces et deux autres phylogénies pour des gènes 1 et 2 étudiés. La présence de deux transferts (entre C et A et entre B et F) rend les deux arbres de gène complètement incongruents. En effet, ils n'ont aucun nœud interne en commun et, donc, aucune histoire évolutive commune. Par conséquent, le mécanisme de transfert horizontal peut complètement altérer la topologie d'un arbre phylogénétique.

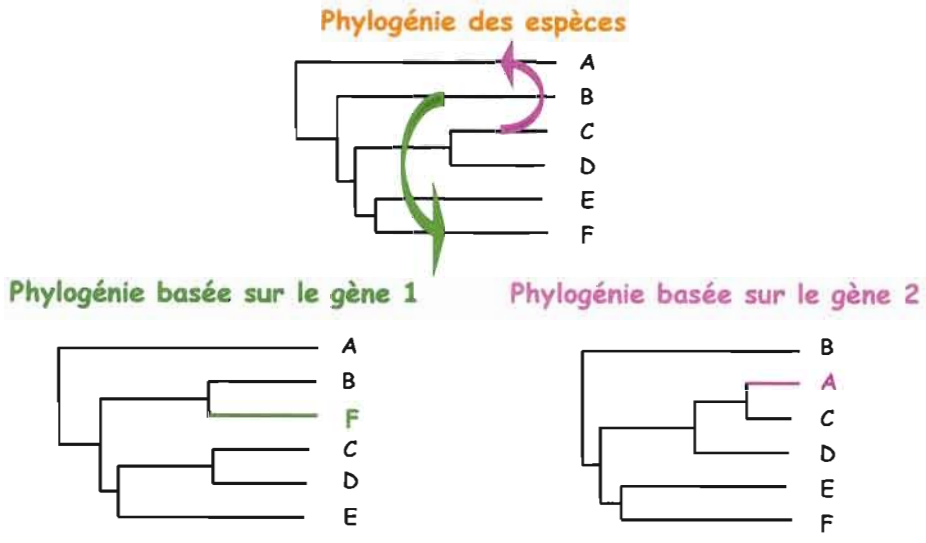


Figure 1.6 Incongruence des arbres de gène par rapport à l'arbre d'espèce, tiré de Philippe *et al.* (2003).

CHAPITRE II

DÉTECTION DES TRANSFERTS HORIZONTAUX DE GÈNES : ÉTAT DE L'ART

2.1 Introduction

Dans ce chapitre, nous présentons une revue de la littérature sur des méthodes et logiciels développés pour la détection et la visualisation de l'évolution réticulée. Après quelques généralités, nous présenterons plus en détail deux applications spécifiques à la détection des transferts horizontaux de gènes, *LatTrans* de Hallett et Lagergren (2001) et *HGT-Detection* (sa première version) de Boc et Makarenkov (2003). Dans notre projet doctoral, nous nous appuierons sur cette première version de l'algorithme *HGT-Detection* et les améliorations qui y ont été apportées dans Makarenkov *et al.* (2006). Enfin, nous présenterons les différents axes qui ont été suivis afin de mener à bien ce projet de recherche.

2.2 L'évolution réticulée

L'évolution réticulée a longtemps été négligée dans les analyses phylogénétiques. Les premières méthodes qui étudiaient les mécanismes de l'évolution réticulée sont apparues au milieu des années 70 (Sneath *et al.*, 1975; Sonea et Panisset, 1976). Plusieurs méthodes ont été proposées pour identifier l'évolution réticulée dans les séquences de nucléotides. Celles-ci incluent : l'affichage de compatibilités (Sneath *et al.*, 1975), des tests de regroupement (Stephens 1985), une approche de randomisation (Sawyer 1989) et une extension de la méthode de reconstruction d'arbres par parcimonie qui permet la recombinaison (Hein 1993). Rieseberg et Morefield (1995) ont développé un programme, *RETICLAD*, qui permet d'identifier les hybrides, fondé sur l'idée qu'ils combinent les caractères de leurs parents. Cependant, ce programme permet de trouver des réticulations seulement entre les arêtes

terminales d'un arbre. Rieseberg et Ellstrand (1993) ont montré des exemples où le programme semble bien fonctionner. La populaire méthode de décomposition en partitions (split-decomposition) rend possible la représentation des données sous la forme d'un split-graphe révélant les conflits contenus dans les données (Bandelt et Dress, 1992a, 1992b). Dans un split-graphe, une paire de nœuds peut être reliée par un ensemble d'arêtes parallèles décrivant les hypothèses d'évolution alternatives. Une autre interprétation des split-graphes est qu'ils représentent en deux dimensions des similarités entre les espèces étudiées. Hallett et Lagergren (2001) ont montré comment le transfert horizontal de gènes pouvait être détecté en mesurant la différence topologique entre un arbre d'espèces et un arbre de gène. Bryant et Moulton (2002) ont introduit une méthode inférant un réseau, *NeighborNet*, permettant la reconstruction de réseaux phylogénétiques planaires. Chacune de ces méthodes a des propriétés qui la rend utile pour l'analyse de données particulières et elles ont toutes un rôle à jouer dans la détection et la caractérisation de l'évolution réticulée. Legendre et Makarenkov (2002) et Makarenkov et Legendre (2004) ont proposé d'utiliser les réticulogrammes pour détecter les réticulations dans des données évolutives. Ils ont développé une méthode basée sur les distances qui infère des phylogénies réticulées. Cette méthode utilise la topologie d'un arbre phylogénétique comme une structure de base sur laquelle on ajoute, au fur et à mesure et suivant un critère d'optimisation, des arêtes de réticulation pour construire un réticulogramme. Parmi d'autres techniques d'inférence de réseaux phylogénétiques mentionnons : la parcimonie statistique (Templeton *et al.*,1992), le *Netting* (Fitch 1997), les réseaux médians (Bandelt *et al.*,1995 et 2000), les réseaux medians-joints (Foulds *et al.*,1979; Bandelt *et al.*,1999), la parcimonie à variance moléculaire (Excoffier et Smouse, 1994), les pyramides (Diday et Bertrand, 1986) et les hiérarchies faibles (Bandelt et Dress, 1989).

2.3 Les méthodes de détection de transferts horizontaux de gènes

Il existe deux principales approches pour identifier les gènes qui ont été transférés horizontalement. Premièrement, l'analyse du génome de l'hôte peut révéler des séquences avec un taux de GC anormal (Lawrence et Ochman, 1997). En supposant que ces séquences ne soient pas apparues par un processus de sélection, on peut en déduire qu'elles sont le résultat d'un transfert horizontal de gènes. Deuxièmement, la comparaison d'un arbre phylogénétique d'espèces avec une phylogénie basée sur un gène observé, et inférée pour le

même ensemble d'espèces, peut révéler des conflits topologiques qui sont explicables par des transferts horizontaux. Ces arbres d'espèces peuvent être basés sur la morphologie d'espèces ou sur des séquences génétiques réfractaires aux transferts horizontaux (e.g., ARNr 16S ou ARNr 23S). Les gènes ribosomiques peuvent être soumis au THG, mais à un taux relativement faible et peuvent donc souvent être utilisés comme une bonne approximation d'une phylogénie d'espèces en l'absence d'autres données (Acinas *et al.*, 2004).

Cette dernière approche a donné lieu à de nombreuses méthodes qui ont commencé à paraitre au début des années 90. Les premières méthodes, utilisant des modèles basés sur des réseaux, ont été proposées par Hein (1990), von Haeseler et Churchill (1993), Page (1994) et Charleston (1998). Mirkin *et al.* (1995) ont mis en avant une méthode de réconciliation d'arbres qui combine différents arbres de gènes dans une unique phylogénie d'organismes. L'article de Moret *et al.* (2004) présente un survol de la modélisation des réseaux phylogénétiques. Maddison (1997), puis, Page et Charleston (1998), ont décrit un ensemble de règles d'évolution qui doivent être prise en compte quand on modélise les transferts horizontaux de gènes.

Plusieurs méthodes proposées récemment utilisent l'approximation de la distance SPR (Subtree Prune and Regraft) qui est étroitement liée à l'inférence des THG. Cette distance donne le nombre de tranferts dans le scénario le plus parcimonieux (Beiko et Hamilton, 2006). Cependant, Bordewich et Semple (2004) ont montré que calculer la distance SPR entre deux arbres binaires enracinés est un problème NP-difficile.

2.3.1 L'algorithme *LatTrans*

Hallett et Lagergren (2001) et Addario-Berry *et al.* (2003) ont développé un modèle de transfert horizontal de gènes qui compare l'évolution d'un ensemble d'arbres de gènes et celle d'un arbre d'espèces. L'algorithme *LatTrans*, implémentant ce modèle, génère tous les scénarios avec une distance SPR minimale, ce qui est exponentiel sur le nombre de transferts. *LatTrans* procède par la réconciliation des arbres de gènes et d'un arbre d'espèces. Un certain nombre de contraintes ont été introduites dans le modèle pour rendre cette réconciliation biologiquement viable. Si une copie multiple d'un gène apparaît dans l'arbre d'espèces, l'algorithme l'interprète comme un possible transfert horizontal. Le scénario de transferts

horizontaux du gène *rbcL* inféré par *LatTrans* est présenté sur figure 2.1. Le modèle de Hallett et Lagergren (2001) inclut aussi un paramètre d'activité α qui définit le nombre de gènes autorisés à être actif en même temps.

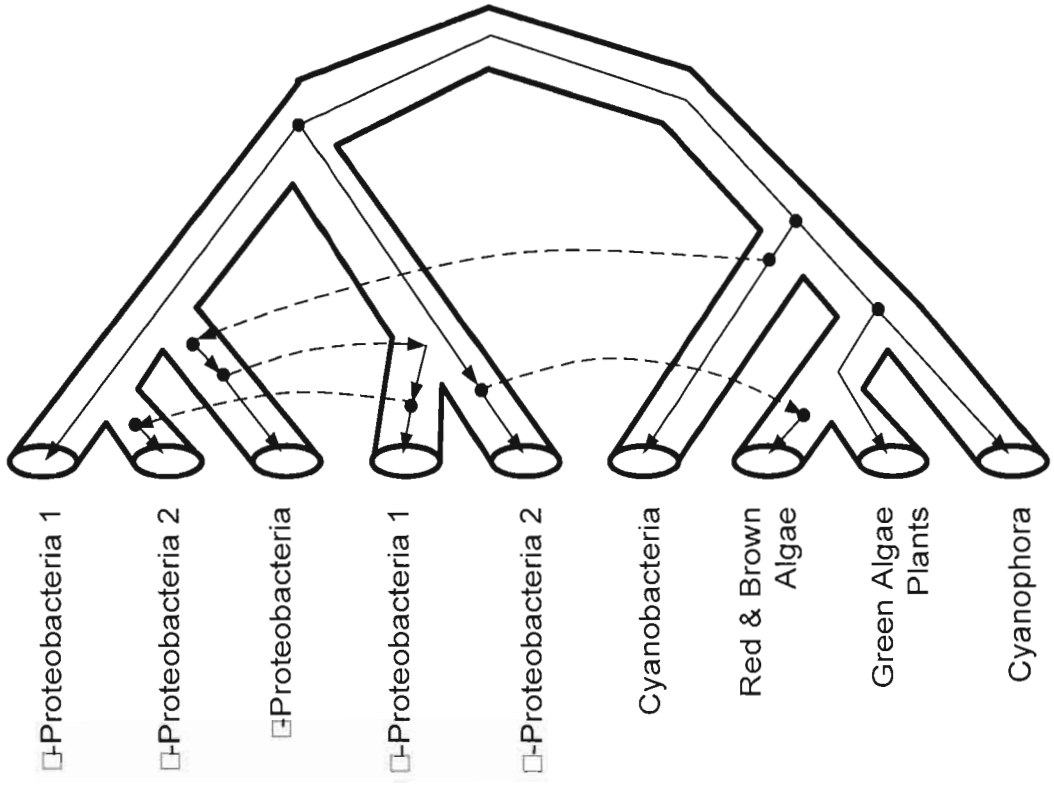


Figure 2.1 Scénario de transferts du gène *rbcL* identifié par Hallett et Lagergren (2001).

Cet algorithme est disponible à l'URL : http://cgm.cs.mcgill.ca/~laddar/lattrans/. Le programme en question inclut aussi une option permettant la recherche de scénarios selon une combinaison de modèles de transferts horizontaux et de duplication de gènes.

2.3.2 L'algorithme *HGT-Detection*

Dans Boc et Makarenkov (2003), nous avons présenté une approche permettant de détecter un scénario de THG en se basant sur la réconciliation métrique ou topologique d'un arbre d'espèces et d'un arbre de gène. Les améliorations de cette méthode ont été décrites dans Makarenkov *et al.* (2006). Nous avions alors présenté deux modèles d'évolution génétique

supposant le transfert d'une partie du gène (Modèle 1) et le transfert du gène au complet (Modèle 2). Nous avions supposé que les arbres phylogénétiques représentant l'évolution du gène considéré et celle des d'espèces ont déjà été construits (à l'aide de la méthode NJ de Saitou et Nei, 1987, par exemple).

2.3.2.1 Description générale

La figure 2.2 ci-dessous illustre les deux modèles considérés. Dans le premier cas, quand une partie du gène transféré du donneur est récupérée par l'espèce hôte, l'arbre phylogénétique d'espèces est transformé en réseau phylogénétique par l'ajout d'une arête orientée représentant le transfert du gène (figure 2.2, l'arbre du bas à gauche). Le deuxième modèle d'évolution considéré suppose que suite au transfert du gène au complet, l'arbre phylogénétique est transformé en un autre arbre phylogénétique (figure 2.2, l'arbre du bas à droite).

2.3.2.2 Modèle 1 : Transfert partiel

Sur la figure 2.2 (l'arbre du haut), l'arête en pointillé reliant les arêtes (3,4) et (2,C) représente un transfert horizontal de gène. Dans un arbre phylogénétique, il existe toujours un chemin unique reliant toute paire de sommets. Si le modèle supposant un transfert partiel est considéré, l'addition d'une arête représentant un transfert horizontal crée un autre chemin entre certains sommets, en transformant l'arbre phylogénétique en réseau (figure 2.2, l'arbre du bas à gauche). Comme le gène qui continue d'évoluer vers l'espèce C comporte maintenant une partie du gène transféré de l'arête (3,4) et une partie originale provenant de l'espèce 2, la distance d'évolution entre l'espèce C et toute autre espèce dans ce graphe doit être calculée comme la somme des parties des longueurs des chemins liant ces deux espèces. Ces chemins passent donc par l'arête (2,6) et par l'arête (7,6), voir figure 2.2 (l'arbre du bas à gauche). Plusieurs règles d'évolution ont été incorporées dans ce modèle pour permettre une meilleure interprétation biologique. Parmi ces règles, nous avons : la définition du sens de l'évolution – de la racine vers les feuilles, l'interdiction de transferts entre les arêtes situées sur la même lignée, l'interdiction de plusieurs transferts croisés entre deux lignées données, etc. (Boc *et al.*, 2004; Makarenkov *et al.*, 2004; Boc *et al.* 2010a). Ce modèle est plus

générique que le Modèle 2, qui suppose que le gène transféré du donneur supplante au complet le gène homologue de l'espèce hôte.

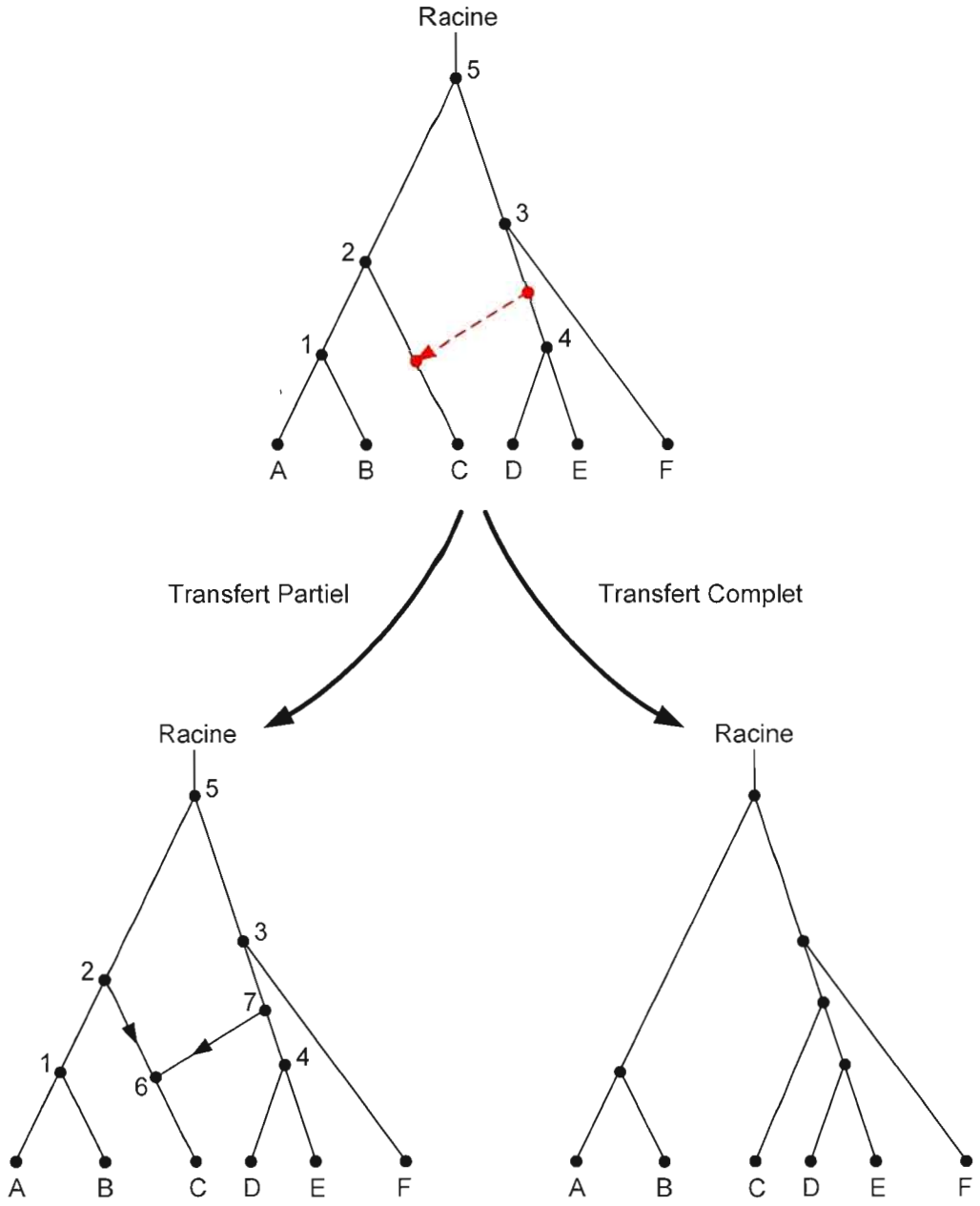


Figure 2.2 Comparaison des deux modèles de transferts horizontaux.

Cependant, l'incorporation de nombreuses règles biologiques et la transformation de l'arbre phylogénétique en réseau font en sorte que le calcul des chemins dans le graphe orienté obtenu après l'ajout des arêtes de transferts horizontaux doit se faire par approximation (Makarenkov *et al.*, 2006). L'utilisation des formules d'approximation permet d'effectuer le calcul en temps polynomial. L'algorithme réalisant ce modèle utilise l'optimisation par les moindres carrés (Gauss, 1811) pour déterminer les transferts de gènes les plus appropriés sous les contraintes biologiques définies.

Optimisation par les moindres carrés

La figure 2.3 illustre un modèle général du transfert partiel. Le chemin de poids minimum entre les taxons i et j changera après l'ajout d'une nouvelle arête orientée (a,b), dirigée de b vers a. D'un point de vue biologique, il est plausible de considérer que le transfert horizontal de gène entre b et a affecte la distance évolutive entre le taxon i et le taxon j, dont la position dans l'arbre phylogénétique est fixe, si et seulement si i est situé au dessous de l'arête de transfert. Par contre, la distance évolutive entre i_1 et j ne sera pas affectée par ajout de l'arête (a,b).

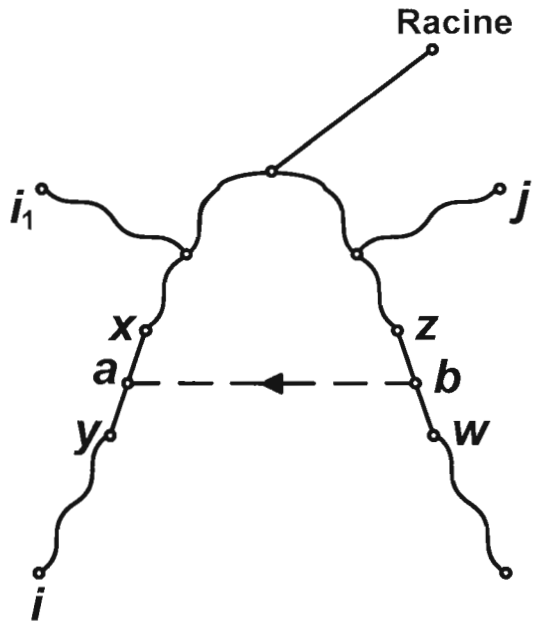


Figure 2.3 Modèle d'évolution impliquant des transferts partiels.

L'ensemble $A(a,b)$ de toutes les paires de taxons telles que la taille minimum du chemin entre eux peut changer si un transfert (a,b) est ajouté à l'arbre d'espèces T est trouvé comme suit:

$$Min\{d(i,a) + d(j,b);\ d(j,a) + d(i,b)\} < d(i,j), \qquad (1)$$

où $d(i,j)$ est le chemin de taille minimum entre les taxons i et j dans T. Alors, $A(a,b)$ est l'ensemble de toutes les paires de feuilles ij telles que $dist(i,j) > 0$; $dist(i,j)$ indique la distance entre les nœuds a et b avant l'ajout du transfert. La fonction des moindres carrés à minimiser, avec l comme variable inconnue, est de la forme suivante :

$$Q(ab,l) = \sum_{dist(i,j)>l} (Min\{d(i,a) + d(j,b);\ d(j,a) + d(i,b)\} + l - \delta(i,j))^2 + \sum_{dist(i,j)\leq l} (d(i,j) - \delta(i,j))^2, \qquad (2)$$

où $\delta(i,j)$ est le chemin de poids minimum entre les taxons i et j dans l'arbre de gène T_1. La fonction $Q(ab,l)$ mesure le gain en ajustement quand l'arête (a,b) de longueur l est ajoutée à l'arbre d'espèces T. La complexité de cet algorithme est $O(kn^4)$ pour produire un scénario de THG avec k transferts pour n taxons.

2.3.2.3 Modèle 2 : Transfert complet

Les transferts horizontaux sont ajoutés dans l'arbre phylogénétique d'espèces en le transformant ainsi en arbre phylogénétique du gène donné. Deux critères d'optimisation, topologique et métrique, ont été utilisés dans le calcul pour déterminer les transferts horizontaux les plus appropriés (Boc et Makarenkov, 2003). Nous avons considéré la distance de Robinson et Foulds (Robinson et Foulds, 1981), comme critère topologique, et les moindres carrés, comme critère métrique. À la première itération, le transfert diminuant le plus la distance de Robinson et Foulds entre l'arbre du gène et celui d'espèces est considéré comme le plus approprié. L'arête du transfert est par la suite ajoutée à l'arbre phylogénétique d'espèces et ainsi de suite. Comme dans ce modèle nous ne considérons que le transfert du gène au complet, l'arête reliant l'espèce affectée par ce transfert et son ancêtre direct est supprimée de l'arbre. Puisque l'ajout d'une arête de transfert est suivi par la suppression d'une arête, nous travaillons toujours avec un graphe connexe et sans cycles, qui est donc un arbre. Cette méthode nécessite également $O(kn^4)$ opérations pour ajouter k transferts dans

l'arbre phylogénétique à n espèces pour les deux critères – topologique et métrique. Le premier d'entre eux est le critère des moindres carrés (LS) défini par la fonction Q suivante :

$$Q = \sum_i \sum_j (d(i,j) - \delta(i,j))^2 \tag{3}$$

où $d(i,j)$ est la distance entre les espèces i et j dans l'arbre phylogénétique d'espèces T et $\delta(i,j)$ est la distance entre les espèces i et j dans l'arbre de gène T' (construit pour le même ensemble d'espèces).

Le second critère qui mesure la différence entre les phylogénies d'espèces et de gène est la distance topologique de Robinson et Foulds (1981). La distance de Robinson et Foulds (RF) représente le nombre minimum d'opérations élémentaires de contraction et d'expansion de nœuds nécessaires pour transformer un arbre phylogénétique en un autre. La figure 2.4 ci-dessous illustre les deux opérations nécessaires pour transformer l'arbre T en l'arbre T_1.

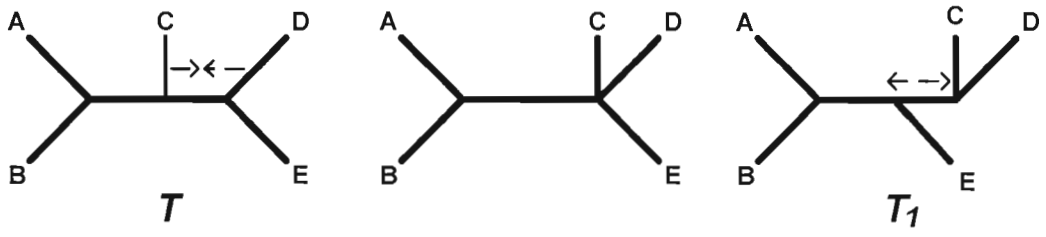


Figure 2.4 La distance de Robinson et Foulds entre T et T_1 est égale à 2.

Quand la distance RF est considérée, nous pouvons l'utiliser comme un critère d'optimisation comme suit : toutes les transformations possibles de l'arbre d'espèces, consistant au transfert d'un de ses sous-arbres d'une arête vers une autre, sont évaluées en calculant la distance RF entre l'arbre d'espèces transformé T_1 et l'arbre de gène T'. Le transfert de sous-arbres produisant la distance RF minimale est retenu. Notons que le problème demandant de trouver le nombre minimum de transferts de sous-arbres nécessaires pour transformer un arbre en un autre, aussi connu comme le problème de transfert des sous-arbres, a été montré NP-complet (Hein *et al.*, 1996).

Les modèles décrits dans Boc et Makarenkov (2003) ont été implémentés dans le logiciel T-Rex (Makarenkov, 2001[1]). La méthode de détection du transfert complet améliorée décrit dans Makarenkov *et al.* (2006) a été implémentée dans la version Web de T-Rex et est accessible à l'adresse suivante : http://www.trex.uqam.ca. La figure 2.6 montre un exemple de résultat obtenu où deux transferts horizontaux sont détectés et incorporés dans une phylogénie de 40 espèces. Plusieurs exemples d'interfaces de la version Web de T-Rex sont présentés en Annexe A.

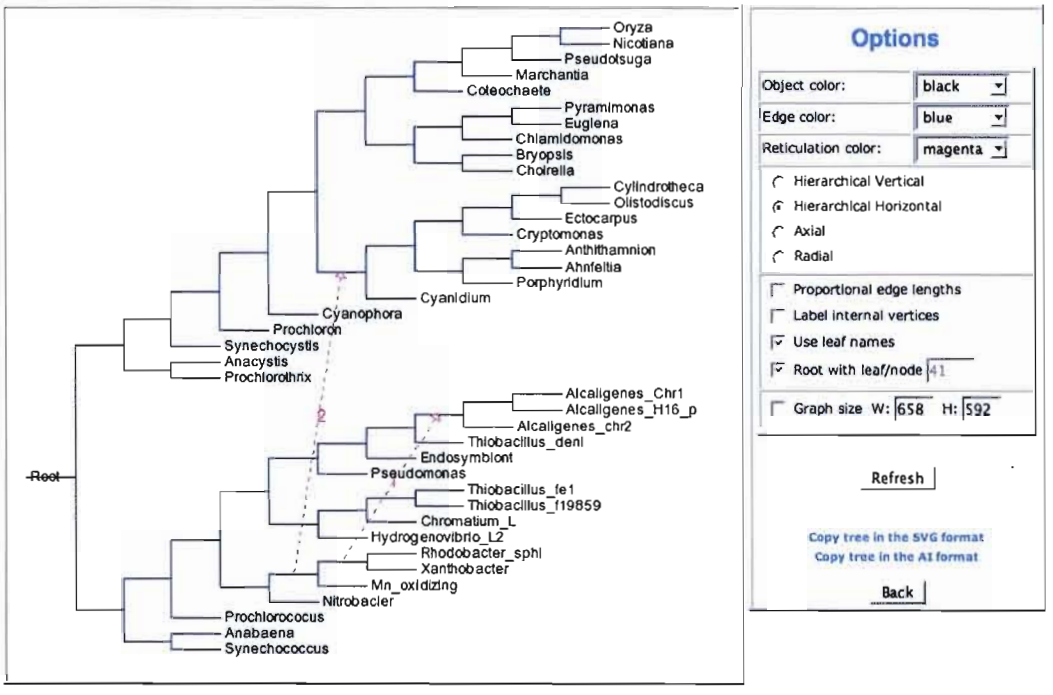


Figure 2.5 Un résultat du programme *HGT-Detection* inclus dans la version Web de T-Rex.

[1] Même si la publication originale sur T-Rex date de 2001, nous y avons ajouté de nouvelles méthodes que nous avons développées plus tard.

2.3.3 Autres travaux dans le domaine des transferts horizontaux

Mirkin *et al.* (2003) ont conçu un algorithme pour la réconciliation de modèles phylétiques avec un arbre d'espèces, en postulant la perte, l'émergence et le transfert de gènes. Ces auteurs ont montré que dans chaque situation, leur algorithme fournit un scénario d'évolution parcimonieux consistant à représenter dans un arbre phylogénétique d'espèces la perte et le gain de gènes. Hallett *et al.* (2004) ont introduit un modèle combinatoire incorporant des THG et des duplications. L'algorithme *HorizStory*, destiné à approximer la distance SPR entre des arbres phylogénétiques enracinés, et possiblement non binaires, a été décrit par MacLeod *et al.* (2005). Cet algorithme commence par éliminer des sous-arbres identiques dans les arbres de gène et d'espèces. Les opérations SPR sont ensuite exécutées récursivement jusqu'à la réconciliation complète des deux arbres. Beiko et Hamilton (2006) ont présenté l'algorithme *Efficient Evaluation of Edit Paths* (*EEEP*) recherchant un nombre minimum de transferts SPR entre deux arbres enracinés. L'approche adoptée par *EEEP* considère les bipartitions induites par les arêtes de l'arbre de référence et de l'arbre de test. La clé des comparaisons topologiques dans cet algorithme est la subdivision des bipartitions de l'arbre de référence dans celles qui sont concordantes et discordantes dans l'arbre de test. D'autre part, Nakhleh *et al.* (2005) ont développé l'heuristique *RIATA-HGT* basée sur une approche diviser pour régner. Than et Nakhleh (2008) ont montré que la dernière version de *RIATA-HGT* est considérablement plus rapide que *LatTrans*, alors que les deux algorithmes sont presqu'équivalents en termes de précision. Récemment, les premiers modèles probabilistes et parcimonieux des THG ont fait leur apparition. Csürös and Miklós (2006) ont introduit un modèle d'évolution de Markov d'une famille de gènes dans un arbre phylogénétique. Ce modèle inclut des paramètres pour le taux de THG, de duplication et de perte de gènes. Jin *et al.* (2006 et 2007) ont décrit deux nouveaux algorithmes pour inférer des THG, dans un cadre comprenant, respectivement, des modèles de maximum de parcimonie et de maximum de vraisemblance.

Bien que toutes ces approches proposent des résultats exploitables, elles sont très sensibles aux types de données utilisées et aucunes d'elles n'apportent une solution complète, incluant la validation statistique des transferts obtenus, à la résolution du problème de détection des THG. Nous avons apporté une première pierre à l'édifice, en proposant une

procédure de validation des transferts obtenus, ainsi qu'un nouveau modèle de détection des transferts partiels. Ce dernier modèle est pour le moment très limité et demande à être amélioré d'un point de vue de la complexité algorithmique et de la viabilité biologique.

2.4 Les objectifs du projet doctoral

Dans le cadre de cette thèse, nous nous posons les objectifs suivants : le premier est de proposer un modèle robuste d'évolution prenant en compte le transfert horizontal de gènes. Ce modèle se basera sur la réconciliation des phylogénies d'espèces et de gène, tout en incorporant des règles d'évolution nécessaires. Le deuxième objectif est de proposer un algorithme général permettant de procéder à la détection des transferts horizontaux de gènes, en se basant sur le modèle d'évolution défini. Cet objectif sera réalisé en trois étapes.

Étape 1 : Développer un algorithme permettant la détection d'un scénario de transferts complets d'un gène. Nous avons identifié les points suivants comme étant les caractéristiques importantes d'un tel algorithme :

1) Complexité algorithmique réduite : en maintenant la complexité polynomiale, on s'assure de pouvoir exécuter l'algorithme dans un temps raisonnable et ainsi d'effectuer des études à grande échelle.

2) Viabilité biologique des transferts détectés : un transfert horizontal de gènes est un mécanisme naturel qui répond à certaines règles d'évolution. Chaque THG proposé par notre algorithme doit respecter ces règles.

3) Validation des THG détectés : chaque transfert horizontal détecté doit être accompagné d'un score de bootstrap indiquant sa fiabilité. Ce point est particulièrement important, car il est inconcevable aujourd'hui de générer des solutions sans y associer un taux de confiance.

Étape 2 : Généraliser cet algorithme pour considérer des transferts partiels d'un gène. Cette méthode sera aussi applicable à des génomes complets ou à des ensembles de gènes.

Étape 3 : Appliquer les méthodes développées à des domaines connexes, par exemple pour modéliser des échanges de mots en biolinguistique.

CHAPITRE III

ALGORITHME POUR LA DÉTECTION DES TRANSFERTS HORIZONTAUX DE GÈNES COMPLETS

3.1 Introduction

Dans ce chapitre, nous décrivons un nouvel algorithme efficace pour l'inférence et la validation des transferts horizontaux de gènes développé dans le cadre du Modèle 1 (transferts complets). Tout d'abord, nous allons introduire et étudier une nouvelle mesure de comparaison entre deux phylogénies : la dissimilarité de bipartitions (Makarenkov et al., 2007; Boc et al., 2010a). Cette mesure de proximité entre deux arbres phylogénétiques peut être considérée comme un raffinement de la distance RF (Robinson et Foulds, 1981) qui prend en compte seulement les bipartitions identiques entre les phylogénies comparées. Nous montrerons que l'utilisation de la dissimilarité de bipartitions (BD) comme critère d'optimisation offre d'importantes améliorations sur les mesures bien connues telles que les moindres carrés (LS), la distance de Robinson et Foulds (RF) et la distance de quartets (QD). Nous décrirons alors l'algorithme proprement dit de détection des THG complets. Par la suite, une procédure de validation évaluant la fiabilité des transferts obtenus sera présentée, puis une comparaison des performances du nouvel algorithme utilisant le critère d'optimisation BD avec celles de *LatTrans* (Hallett et Lagergren, 2001) et *RIATA-HGT* (Nakhleh *et al.*, 2005; Than et Nakhleh, 2008) sera effectuée. Ces trois algorithmes seront comparés en termes de taux de transferts détectés et de temps d'exécution. Enfin, notre algorithme sera appliqué à deux jeux de données réelles, le premier, traitant de l'évolution du gène *rpl12e* originalement considéré par Matte-Tailliez *et al.* (2002) et le second, étudiant l'évolution des séquences *PheRS* originalement considéré par Woese *et al.* (2000).

3.2 Nouvel algorithme pour l'inférence et la validation des THG

Le nouvel algorithme pour la détection des transferts horizontaux de gènes procède par une réconciliation progressive d'une phylogénie d'espèces enracinée et d'une phylogénie d'un gène, notées respectivement T et T'. À chaque étape de l'algorithme, plusieurs paires d'arêtes de T sont testées pour évaluer l'hypothèse qu'un THG se soit produit entre elles. Le modèle de THG fonctionne dans les deux cas suivant : (1) quand le gène transféré supplante le gène orthologue du génome receveur ; et (2) quand le gène transféré, absent du génome receveur, y est ajouté.

L'arbre phylogénétique d'espèces original T est graduellement transformé en l'arbre phylogénétique du gène T' par une série d'opérations SPR (i.e., THG). Le but étant de trouver la plus petite séquence possible d'arbres T, T_1, T_2, ..., T' qui transforme T en T'. Un certain nombre de contraintes d'évolution doit être pris en compte car un THG exige que les espèces source et destination soient contemporaines.

Par exemple, le transfert sur une même lignée (figure 3.1a) et les transferts qui se croisent tel qu'illustré à la figure 3.1b-d, conduisent à des scénarios inappropriés et doivent être interdits (voir aussi Maddison, 1997 ; Page et Charleston, 1998 ou Hallett et Lagergren, 2001).

Le problème du calcul de la distance SPR est connu comme étant NP-difficile pour les arbres enracinés et non-enracinés. La première preuve de cette complexité, dans le cas des arbres non-enracinés a été donnée par Hein *et al.* (1996). Cependant, Allen et Steel (2001) l'ont trouvé incorrecte. Ils ont montré que le problème similaire de la distance TBR (Tree Bisection and Reconnection) était aussi NP-difficile. Hickey *et al.* (2006) ont alors produit une preuve complète pour les arbres non-enracinés. Par ailleurs, Bordewich et Semple (2004) ont montré que calculer la distance SPR entre des arbres binaires enracinés ou non-enracinés est un problème NP-difficile.

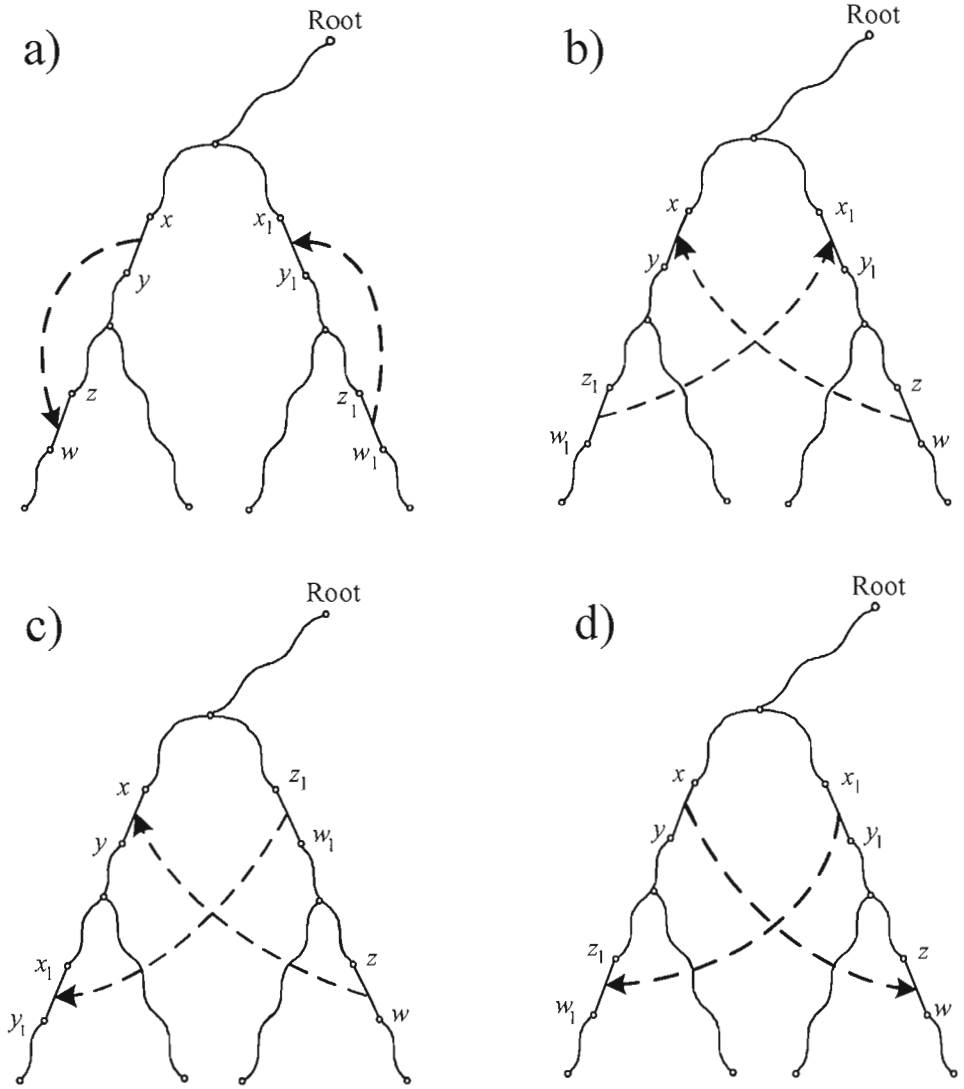


Figure 3.1 Cas de figures où un transfert de gène est interdit. Cas a : les transferts entre les branches situées sur la même lignée doivent être interdits. Cas b, c et d : les transferts croisés comme suit, doivent aussi être interdits. Une branche est représentée par une ligne droite et un chemin par une ligne ondulée.

3.2.1 La dissimilarité de bipartitions et autres critères d'optimisation

Nous considérons quatre critères d'optimisation qui peuvent être utilisés pour sélectionner les meilleurs THG à chaque étape de l'algorithme. Le premier d'entre eux est le critère des moindres carrés (voir la formule 3 au chapitre II).

Le second critère qui peut être utilisé pour évaluer l'écart entre les phylogénies d'espèces et de gène est la distance topologique de Robinson et Foulds (RF) (Robinson et Foulds, 1981). Le troisième critère considéré, la distance de quartets (QD), est le nombre des quartets, sous-arbres induits par quatre feuilles, qui diffèrent entre les arbres comparés. Nous pouvons utiliser ces critères comme suit pour déterminer le meilleur THG à chaque pas de l'algorithme. Quand plusieurs transformations de l'arbre d'espèces, consistant en des opérations SPR entre ses sous-arbres, sont évaluées, celle qui fournit la valeur minimale du critère sélectionné mesuré pour l'arbre d'espèces transformé T_1 et l'arbre de gène T' est retenue.

Le quatrième critère d'optimisation, la *dissimilarité de bipartitions* (Makarenkov et al., 2007 ; Boc et al., 2010a), est défini comme suit. Sans perte de généralité, nous assumons que T et T' sont des arbres phylogénétiques binaires avec le même ensemble de feuilles, c'est-à-dire, espèces ou taxons. Un vecteur de bipartitions d'un arbre T est un vecteur binaire induit par une arête de T. Soit ***BT*** la table de bipartitions des arêtes internes de l'arbre T (c'est-à-dire, la table incluant tous les vecteurs de bipartitions induits par les arêtes internes de T) et ***BT'*** la table de bipartitions des arêtes internes de l'arbre T'. La dissimilarité de bipartitions *bd* entre T et T' est calculée comme suit :

$$bd = \Big(\sum_{a \in BT} \underset{b \in BT'}{Min}(Min\,(d(a,b);d(a,\bar{b}))) + \sum_{b \in BT'} \underset{a \in BT}{Min}\,(Min\,(d(b,a);d(b,\bar{a})))\Big)/2, \quad (1)$$

où $d(a,b)$ est la distance de Hamming (Hamming, 1950) entre les vecteurs de bipartitions a et b, et $\bar{a}$ et $\bar{b}$ sont les compléments de a et b, respectivement. Une telle mesure représente un raffinement de la mesure RF qui prend en compte seulement les bipartitions identiques. Par exemple, la dissimilarité de bipartitions entre les arbres T et T' avec 6 feuilles (figure 3.2) est calculée comme suit : $bd(T,T')= ((2 + 1 + 2) + (2 + 1 + 1))/2 = 4,5$. Ici, la distance de

Hamming minimale entre les bipartitions (directe et son complément) correspondantes à l'arête *a* et tous les vecteurs de la table de bipartitions ***BT'*** est 2. C'est la distance entre les vecteur *a* et $\overline{f}$, et *a* et *d* (dans la figure 3.2, seule l'association entre *a* et $\overline{f}$ est représentée par une flèche). Pour la bipartition *b*, cette distance est 1 (la distance entre *b* et *d*) et pour la bipartition *c*, cette distance est 2 (la distance entre *c* et *d*). De la même façon, la distance minimale entre la bipartition *e* et toutes les bipartitions de ***BT*** est 2 (avec $\overline{b}$), pour la bipartition *f* cette distance est 1 (avec $\overline{b}$ aussi) et pour la bipartition *d*, elle est égale à 1 (avec *b*).

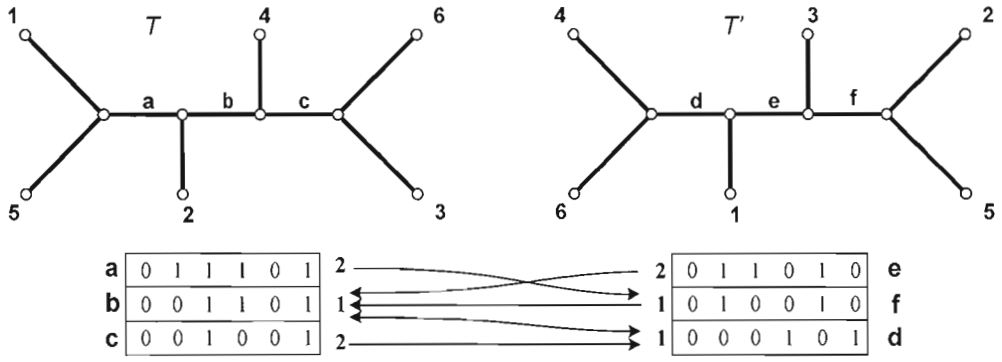


Figure 3.2 Les arbres *T* et *T'* et leur table de bipartitions. Chaque ligne de la table de bipartitions correspond à une branche interne de l'arbre. Les flèches indiquent les associations entre les vecteurs de bipartitions dans les deux tables. La valeur en gras proche de chaque vecteur représente la distance associée.

Cet exemple montre que plusieurs vecteurs de bipartitions de la première table de bipartitions peuvent être associés au même vecteur de bipartitions de la seconde table, par exemple, *d*, *e* et *f* sont associés à *b* (ou $\overline{b}$), et *b* et *c* sont associés à *d* (figure 3.2). Il est important de mentionner que la dissimilarité de bipartitions n'est pas toujours une distance (i.e., une métrique). Pour les arbres avec cinq feuilles et plus, il est possible d'exhiber trois topologies d'arbres pour lesquelles l'inégalité triangulaire n'est pas respectée.

Les propositions 1 et 2 ci-dessous établissent quelques propriétés intéressantes de la dissimilarité de bipartitions (Makarenkov et al., 2007; Boc et al., 2010a). Ainsi, la proposition

1 énonce la condition suffisante assurant qu'une dissimilarité de bipartitions (BD) satisfasse l'inégalité triangulaire (et soit donc une distance), alors que la proposition 2 établit la valeur maximale de BD en fonction du nombre de feuilles. La bipartition a d'un arbre T est associée à la bipartition b d'un arbre T' (cette association est notée $a \rightarrow b$), si la distance de Hamming entre les vecteurs de bipartitions correspondant à a et à b est la plus petite de toutes les distances possibles calculées entre tous les vecteurs de bipartitions de T'. □

Proposition 1. *Soit T_1, T_2 et T_3 des arbres phylogénétiques avec le même nombre d'arêtes internes et le même ensemble de feuilles. Si pour chaque paires de bipartitions a et b d' arbres différents : $a \rightarrow b$ implique que $b \rightarrow a$, et que pour chaque triplet de bipartitions $a \in T_1$, $b \in T_2$ et $c \in T_3$: $a \rightarrow b$ et $b \rightarrow c$ implique que $a \rightarrow c$, alors l'inégalité triangulaire $bd(T_1, T_2) \leq bd(T_1, T_3) + bd(T_2, T_3)$ est satisfaite.*

Preuve. D'une part, en considérant la première partie de l'énoncé de la proposition :

$$bd(T_1,T_2) = \Big(\sum_{\substack{a \in BT_1, \\ (b \in BT_2) \rightarrow a}} d(a,b) + \sum_{\substack{b \in BT_2, \\ (a \in BT_1) \rightarrow b}} d(b,a) \Big)/2 = \sum_{\substack{a \in BT_1, \\ (b \in BT_2) \rightarrow a}} d(a,b),$$

où ($a \in BT_1$ et $b \in BT_2 \rightarrow a$) signifie que la somme est prise pour tous les vecteurs a de la table de bipartitions BT_1 correspondant à l'arbre T_1 et tous les vecteurs b associés à ces vecteurs a. De façon similaire :

$$bd(T_1,T_3) = \Big(\sum_{\substack{a \in BT_1, \\ (c \in BT_3) \rightarrow a}} d(a,c) + \sum_{\substack{b \in BT_3, \\ (c \in BT_1) \rightarrow a}} d(c,a) \Big)/2 = \sum_{\substack{a \in BT_1, \\ (c \in BT_3) \rightarrow a}} d(a,c), \text{ et}$$

$$bd(T_2,T_3) = \Big(\sum_{\substack{b \in BT_2, \\ (c \in BT_3) \rightarrow b}} d(b,c) + \sum_{\substack{c \in BT_3, \\ (b \in BT_2) \rightarrow c}} d(c,b) \Big)/2 = \sum_{\substack{b \in BT_2, \\ (c \in BT_3) \rightarrow b}} d(b,c).$$

Considérons les trois sommes suivantes : $\sum_{\substack{a \in BT_1, \\ (b \in BT_2) \rightarrow a}} d(a,b)$, $\sum_{\substack{a \in BT_1, \\ (c \in BT_3) \rightarrow a}} d(a,c)$ et $\sum_{\substack{b \in BT_2, \\ (c \in BT_3) \rightarrow b}} d(b,c)$.

Comme la distance de Hamming, d, satisfait l'inégalité triangulaire, pour chaque terme $d(a,b)$

de la première somme, nous avons le terme $d(a,c)$ de la seconde somme et le terme $d(b,c)$ de la troisième somme tels que : $d(a,b) \leq d(a,c) + d(b,c)$. Comme chacun des vecteurs de bipartitions inclus dans les tables de bipartitions BT_1, BT_2 et BT_3 n'apparait seulement qu'une fois dans chacune des trois sommes, nous concluons que: $bd(T_1,T_2) \leq bd(T_1,T_3) + bd(T_2,T_3)$. □

Proposition 2. *La valeur de la dissimilarité de bipartitions entre deux arbres phylogénétiques définis sur le même ensemble de n feuilles varie entre 0 et n(n-3)/2 si n est pair et entre 0 et (n-1)(n-3)/2 si n est impair.*

Preuve. Pour chaque paire de vecteurs binaires a et b de taille n, la valeur maximale de la quantité $Min\,(d(a,b);d(a,\bar{b}))$, où $d(a,b)$ est la distance de Hamming entre a et b et $\bar{a}$ et $\bar{b}$ sont leurs compléments, est $n/2$ quand n est pair et $(n-1)/2$ quand n est impair. D'autre part, le nombre maximal d'arêtes internes dans un arbre phylogénétique (c'est-à-dire, le nombre de lignes dans la table de bipartitions correspondante) avec n feuilles est $n-3$. Par conséquent, selon la formule 1, la valeur maximale de la dissimilarité de bipartitions entre deux arbres de n feuilles est $n(n-3)/2$ si n est pair et $(n-1)(n-3)/2$ si n est impair. □

3.2.2 La contrainte de sous-arbres

Dans cette section nous discutons d'une des caractéristiques principales du nouvel algorithme d'inférence de transferts horizontaux de gènes. Considérons un THG dans l'arbre d'espèces T, allant de a à b et le transformant en un arbre T_1 (voir la figure 3.3). La contrainte suivante est énoncée : pour permettre le THG entre les arêtes (x,y) et (z,w) de l'arbre d'espèces T, le clade consistant en le sous-arbre enraciné par l'arête (x,a), et incluant les nœuds y et w dans T_1, doit être présent dans l'arbre de gène T' (Makarenkov et al., 2006 ; Boc et al., 2010a).

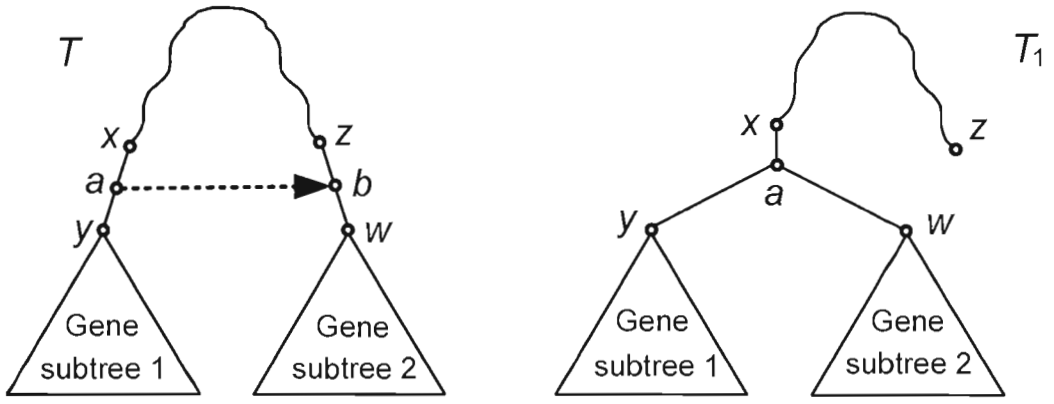


Figure 3.3 Illustration de la contrainte de sous-arbres. Le transfert entre les branches (*x*,*y*) et (*z*,*w*) dans l'arbre d'espèces *T* est permis si et seulement si les sous-arbres enracinés par les branches (*x*,*a*) et (z,*w*), ainsi que leur regroupement enraciné par (*x*,*a*), sont présents dans l'arbre de gène *T'*.

Une telle contrainte, appelée ici, *contrainte de sous-arbres*, nous permet tout d'abord d'arranger les conflits topologiques entre *T* et *T'*, qui sont dus aux transferts entre les ancêtres des espèces contemporaines, qui sont plus facile à détecter, puis d'identifier les transferts apparus plus profondément dans la phylogénie. En outre, l'utilisation de la contrainte de sous-arbres permet de prendre en compte automatiquement toutes les contraintes d'évolution requises (figure 3.1) car les deux sous-arbres impliqués dans le THG doivent être présents dans l'arbre de gènes *T'*, ainsi que le nouveau sous-arbre qu'ils forment après le transfert (figure 3.3). En effet, si les THG sur une même lignée (figure 3.1a) où les transferts croisés présentés sur la figure 3.1(b-d) étaient permis, la contrainte de sous-arbres ne serait pas respectée (le lecteur est renvoyé à la section de discussion où tous les avantages apportés par cette contrainte sont résumés).

Les deux théorèmes suivants établissent quelques propriétés des bipartitions dans le contexte des THG satisfaisant la contrainte de sous-arbres. Ces propriétés sont utilisées dans l'algorithme de détection des transferts horizontaux décrit ci-après.

Théorème 1. *Si le sous-arbre Sub_{yw} nouvellement formé et résultant du THG (à savoir le sous-arbre enraciné par l'arête (x,a) dans la figure 3.3) est présent dans l'arbre de gène T', et que le vecteur de bipartitions associé à l'arête (x,x_1) dans l'arbre d'espèces transformé T_1 (figure 3.4) est présent dans la table de bipartitions de T', alors le THG de (x,y) à (z,w), transformant T en T_1, fait partie d'un* ***scénario de coût minimal*** *qui transforme T en T' et qui satisfait la contrainte de sous-arbres.*

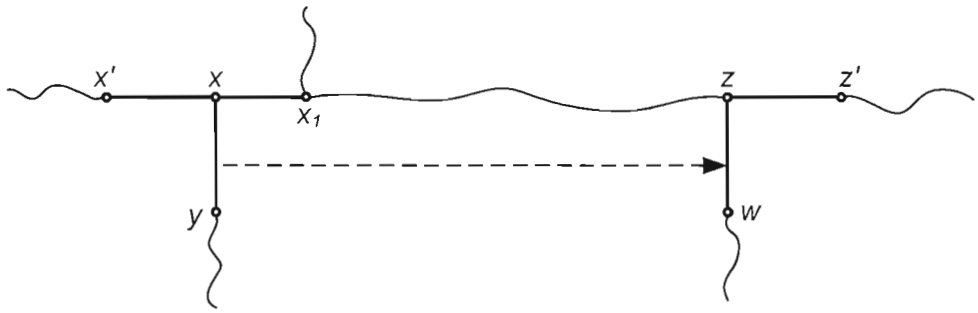


Figure 3.4 Le transfert entre les arêtes (x,y) et (z,y) fait partie du scénario de coût minimal. Un scénario de coût minimal transforme l'arbre d'espèces T en l'arbre de gène T' si la bipartition correspondante à la branche (x,x_1) dans l'arbre transformé T_1 est présente dans la table de bipartitions T' et le sous-arbre Sub_{yw} est présent dans T'; (Théorème 2) tout scénario de coût minimum transforme l'arbre d'espèces T en l'arbre de gène T' si toutes les bipartitions correspondantes aux branches du chemin (x',z') dans l'arbre d'espèces T_1 sont présentes dans la table de bipartitions de T' et le sous-arbre Sub_{yw} est présent dans l'arbre T'.

Preuve. Les quatre cas possibles menant à la formation du sous-arbre Sub_{yw} sont les suivants : 1) Le THG de (x,y) à (z,w); 2) Le THG de (z,w) à (x,y); 3) le THG de (x',x) à (z,z'); 4) le THG de (z,z') à (x',x). Quand le chemin (x,z) dans T comprend au moins deux arêtes, les THG correspondant aux cas (3) et (4) ne produiront pas le sous-arbre Sub_{yw}, mais amèneront les sommets x et z plus proche l'un de l'autre, réduisant ainsi le nombre d'arêtes du chemin (x,z). Les THG (3) et (4) induiront la bipartition b, qui sera présente dans la table de bipartitions de l'arbre de gène T' en raison de la contrainte de sous-arbres, telle que les feuilles du sous-arbre situées à la gauche de x' et celles situées à la droite de z' (figure 3.4) se

trouvent du même côté (e.g., elles sont notées par 1 dans la table de bipartitions de T), alors que les feuilles du sous-arbre localisées en dessous des sommets y et w se trouvent de l'autre côté (e.g., elles sont notées par 0 dans la table de bipartitions de T').

Selon les conditions du théorème, la bipartition correspondant à l'arête (x,x_1) dans l'arbre T_1 obtenu à partir de l'arbre d'espèces initial T suite au THG de (x,y) à (z,w), et notée ici $\boldsymbol{b}_1$, est aussi présente dans la table de bipartitions de T'. Cela signifie que les feuilles situées à la gauche de x' et celles des sous-arbres situés en dessous des sommets y et w se trouvent dans la même partie de la bipartition, alors que les feuilles du sous-arbre situé à la droite de z' se trouvent dans l'autre partie (figure 3.4). Évidemment, les bipartitions $\boldsymbol{b}$ et $\boldsymbol{b}_1$ ne sont pas compatibles (i.e., elles ne peuvent pas être présentes ensemble dans la même table de bipartitions associée à un arbre phylogénétique) ce qui signifie que les THG de (x',x) à (z,z') et de (z,z') à (x',x) sont impossibles. En outre, le THG de (z,w) à (x,y) est possible seulement quand le chemin (x,z) dans T consiste en une arête unique (dans le cas des THG opposés, les transferts de (x,y) à (z,w) et de (z,w) à (x,y) mèneront à la même transformation topologique de T) car ce THG induirait une bipartition, notée ici $\boldsymbol{b}_2$, qui est incompatible avec $\boldsymbol{b}_1$ si le chemin (x,z) dans T contient au moins deux arêtes. En effet, dans $\boldsymbol{b}_2$, les feuilles du sous-arbre situé à la droite de z' et celles des sous-arbres situés en dessous des sommets y et w se trouvent dans la même partie de la bipartition, alors que les feuilles du sous-arbre situé à la gauche de x' se trouve dans l'autre partie. Par conséquent, le THG de (x,y) à (z,w) est nécessaire pour transformer T en T'. La seule exception serait le cas du THG opposé, de (z,w) à (x,y), qui est possible seulement si le chemin (x,z) ne contient qu'une seule arête. Dans ce cas, les THG opposés mèneront à la même transformation topologique et aucun d'entre eux ne fera partie d'un scénario de THG de coût minimal transformant T en T' et respectant la contrainte de sous-arbres. □

Théorème 2. *Si le sous-arbre Sub_{yw} nouvellement formé et résultant du THG (i.e. le sous-arbre enraciné par l'arête (x,a) sur la figure 3.3) est présent dans l'arbre de gène T', et que tous les vecteurs de bipartitions associés aux arêtes du chemin (x',z') dans l'arbre d'espèces transformé T_1 (figure 3.4) sont présents dans la table de bipartitions de T', et que le chemin (x',z') dans T_1 comprend au moins 3 arêtes, alors le THG de (x,y) à (z,w), transformant T en T_1, fait partie de* ***n'importe quel scénario de THG de coût minimal*** *transformant T en T' et satisfaisant la contrainte de sous-arbres.*

Preuve. Les vecteurs de bipartitions correspondant aux arêtes (x',x) et (z,z') de l'arbre d'espèces T_1 obtenu à partir de T après le THG de (x,y) à (z,w) sont aussi présents dans la table de bipartitions de l'arbre d'espèces T et de l'arbre de gène T'. Alors, les quatre cas possibles menant à la formation du sous-arbre Sub_{yw} sont les suivants : 1) un THG de (x,y) à (z,w) ; 2) un THG de (z,w) à (x,y); 3) un THG de (x',x) à (z,z') ; 4) un THG de (z,z') à (x',x). Quand le chemin (x,z) dans T comprend deux arêtes ou plus, les THG correspondant aux cas (3) et (4) ne produiront pas le sous-arbre Sub_{yw}, mais amèneront les sommets x et z plus proches l'un de l'autre en réduisant le nombre d'arêtes du chemin (x,z). Selon l'énoncé du théorème, toutes les bipartitions du chemin non-vide (x,z) dans T_1 obtenues à partir de l'arbre d'espèces initial T après le THG de (x,y) à (z,w) sont aussi présentes dans la table de bipartitions de l'arbre de gène T'. Par conséquent, les feuilles du sous-arbre situé à la gauche de x' et celles des sous-arbres situés en dessous des sommets y et w (figure 3.5) se trouvent dans la partie différente (e.g., elles sont notées par des 1 dans la table de bipartitions de T') de ces bipartitions que les feuilles du sous-arbre situé à la droite de z' (e.g., elles sont notées par des 0 dans la table de bipartitions de T'). Cela signifie qu'il n'y a pas de bipartitions dans T' comprenant toutes les feuilles situées dans les sous-arbres à la gauche de x' et à la droite de z', ni celles des sous-arbres situés en dessous des sommets y et w. Alors, le THG de (x',x) à (z,z'), le cas (3), ainsi que le THG inverse de (z,z') à (x',x), le cas (4), violeront la contrainte de sous-arbres. Évidemment, tout THG des arêtes (x',x) et (z,z') aux arêtes du chemin (x,z) violera aussi la contrainte de sous-arbres.

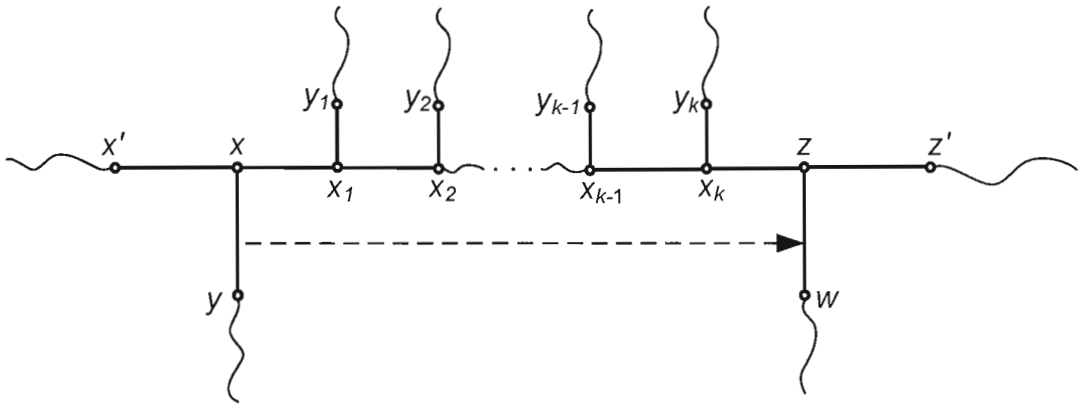


Figure 3.5 Le transfert entre les arêtes (x,y) et (z,w) fait partie du scénario de coût minimal transformant T en T' si toutes les bipartitions des arêtes du chemin (x',z') dans l'arbre d'espèces transformé T_1 sont présentes dans la table de bipartitions de T' et que le sous-arbre Sub_{yw} est présent dans T'.

À cet effet, ni le THG de (x,y) vers (z,w) ni le THG opposé de (z,w) vers (x,y) ne fait partie du scenario de transferts de coût minimal transformant T en T' et satisfaisant la contrainte de sous-arbres. Après le THG allant de (x,y) vers (z,w), tous les vecteurs de bipartitions correspondant aux arêtes du chemin non-vide (x',z'), dans la figure 3.5, seront présents dans la table de bipartitions de T', et aucun d'entre eux dans le cas du THG opposé de (z,w) vers (x,y). Comme les bipartitions associées aux arêtes (x_i,x_{i+1}) et (x_{i+1},x_{i+2}), où $i = 0,\ldots, k\text{-}1$, et $x_0 = x'$ et $x_{k+1} = z'$, (figure 3.5), sont présentes dans la table de bipartitions de T', la bipartition associée aux arêtes (x_{i+1}, y_{i+1}) est aussi présente dans la table de bipartitions de T'. Cela signifie que les sous-arbres enracinés par les arêtes allant de (x_1,y_1) à (x_k,y_k) peuvent être arrangés indépendamment (selon la topologie de l'arbre de gène T'), si ce n'est pas déjà fait, les uns des autres, et du reste de l'arbre T_1 (i.e., les opérations SPR seront effectuées seulement dans les sous-arbres et celles entre les sous-arbres ne seront pas nécessaires). De la même manière, dans un scénario de coût minimal, les arrangements des sous-arbres situés à la gauche de x' et ceux situés à la droite de z' (figure 3.5) doivent êtres faits indépendamment du reste de l'arbre et prendront le même nombre minimal d'opérations SPR dans le cas du THG entre (x,y) et (z,w) et du THG opposé entre (z,w) et (x,y). Par conséquent, dans le cas du THG opposé de (z,w) vers (x,y), la transformation SPR de l'arbre T_1 en l'arbre de gène T'

prendra au moins une opération SPR de plus, nécessaire pour arranger les arêtes du chemin (x,z), que dans le cas du THG de (x,y) à (z,w). □

3.2.3 Algorithme d'inférence des transferts horizontaux de gènes complets

Les principales étapes de l'algorithme, appelé *HGT-Detection*, prévues pour fournir une série de transformations SPR de coût minimal d'un arbre d'espèces donné en un arbre de gène donné sont les suivantes :

Étape préliminaire. Inférons les arbres d'espèces et de gène, notés respectivement par T et T', dont les feuilles sont étiquetées avec le même ensemble de n espèces. Les deux arbres doivent être enracinés en fonction d'évidences biologiques. Si aucune évidence plausible permettant d'enraciner les arbres n'est disponible, les stratégies d'enracinement par *outgroup* (en utilisant un groupe des espèces extérieur au groupe étudié) ou *midpoint* (en utilisant l'arête médiane de l'arbre) peuvent être utilisées. Un enracinement correct des arbres est essentiel car une mauvaise position de la racine dans l'arbre d'espèces ou de gène mènera à l'inférence de transferts étant faux positifs ou faux négatifs. S'il existe des sous-arbres identiques avec au moins deux feuilles appartenant à T et T', la taille du problème peut être réduite en remplaçant ces sous-arbres identiques par la même arête auxiliaire dans T et T'.

Étape k. Considérons tous les transferts possibles entre les paires d'arêtes de l'arbre d'espèces T_{k-1} ($T_0 = T$ à l'étape 1), sauf les transferts entre les paires d'arêtes adjacentes et ceux qui violent la contrainte de sous-arbres.

- Entre tous les THG éligibles, recherchons ceux qui satisfont les conditions du Théorème 2, en premier, et du Théorème 1 en second.

- Effectuons les opérations SPR correspondant à ces THG, qui transforment l'arbre T_{k-1} en l'arbre T_k. Si de tels THG n'existent pas, effectuons toutes les transformations SPR correspondant aux transferts satisfaisant la contrainte de sous-arbres.

- À chaque étape, de multiples opérations SPR (i.e., de multiples THG) peuvent être effectuées. La direction de chaque THG est déterminée en utilisant le critère d'optimisation sélectionné qui peut être dans notre cas : les moindres carrés (LS), la

distance de Robinson et Foulds (RF), la distance de quartets (QD) ou la dissimilarité de bipartitions (BD). Entre deux THG opposés, l'algorithme choisit le transfert qui minimise la valeur du critère d'optimisation sélectionné, calculé pour l'arbre d'espèces transformé et l'arbre de gène T'.

- Réduisons la taille des arbres d'espèces et de gène, en transformant en des arêtes uniques les sous-arbres identiques, nouvellement formés, dans l'arbre d'espèces T_k et dans l'arbre de gène T'.

Condition d'arrêt, complexité algorithmique et procédure d'élimination des transferts inutiles.

La procédure s'arrête quand le coefficient RF, LS, QD ou BD est égal à 0. En raison de la procédure de réduction progressive de la taille des arbres d'espèces et de gène et de la possibilité d'identifier de multiples transferts à chaque étape, la complexité temporelle de l'algorithme proposé est $O(kn^3)$ pour inférer k transferts servant à réconcilier une paire de phylogénies d'espèces et de gène avec n feuilles.

Une fois les arbres d'espèces et de gène réconciliés, une procédure d'élimination des *transferts inutiles* (un transfert inutile, ou redondant, est un transfert dont la suppression ne change pas la topologie de l'arbre de gène résultant) est appliquée. Par exemple, en considérant le scénario de transferts montré à la figure 3.13, le transfert entre *Methanoccocus jannashii* et le clade de cinq taxons, incluant *Archaeoglobus fulgidus*, effectué comme THG numéro 4, serait un transfert inutile (i.e., ce THG serait annulé par les opérations SPR 4 et 5 présentées à la figure 3.13). Si un scénario de k transferts est trouvé par l'algorithme, la procédure d'élimination teste les (k-1) sous-scénarios possibles de transferts tels que dans chacun d'eux un des transferts initialement trouvés est éliminé. Si aucun des (k-1) sous-scénarios ne mène au même arbre de gène, alors la procédure s'arrête sans éliminer de transferts. Autrement, le premier sous-scénario avec (k-1) transferts qui mène au même arbre de gène est retenu, et tous les sous-scénarios avec (k-2) transferts sont testés de la même manière. La procédure s'arrête quand aucun THG inutile ne peut plus être trouvé. La proposition suivante établit deux propriétés intéressantes reliées à la contrainte de sous-arbres.

Le schéma algorithmique complet de *HGT-Detection* est présenté dans l'annexe B.1.

Proposition 3. *Si la contrainte de sous-arbres est appliquée à toutes les étapes de l'algorithme, alors :*

1) L'algorithme de détection a au plus n-3 étapes, et nécessite au plus n-3 transferts (i.e., n-3 opérations SPR), pour transformer un arbre binaire d'espèces T avec n feuilles en un arbre binaire de gène T' défini sur le même ensemble de feuilles.

2) L'arbre de gène T' est toujours retrouvé à la dernière étape de l'algorithme (i.e., T_k=T', en assumant que l'étape k est la dernière étape de l'algorithme) peu importe le critère d'optimisation sélectionné (RF, LS, QD ou BD).

La preuve de cette proposition est basée sur le fait que le nombre maximal d'arêtes internes dans un arbre phylogénétique à n feuilles est n-3, et que chaque opération SPR satisfaisant la contrainte de sous-arbres (Makarenkov et al., 2006 ; Boc et al., 2010a) crée au moins une nouvelle arête interne dans l'arbre d'espèces transformé (e.g., l'arête (x,a) dans la figure 3.3), qui existe déjà dans l'arbre de gène T'. Aussi, tant que les topologies de l'arbre d'espèces transformé et de l'arbre de gène sont différentes, il existe au moins deux opérations SPR (incluant les transferts opposés), satisfaisant la contrainte de sous-arbres, qui peuvent être effectuées. Le lecteur est aussi référé à Bordewich *et al.* (2009 ; Théorèmes 3.1 et 4.1), où les auteurs prouvent l'existence d'une séquence d'opérations SPR, transformant T en T', dans le sens où n'importe quel arbre T_p dans la séquence est obtenu à partir de T_{p-1} par une simple opération SPR, et $RF(T_p,T') < RF(T_{p-1},T')$ ou, respectivement, $QD(T_p,T') < QD(T_{p-1},T')$. □

Même si la contrainte de sous-arbres n'est pas formulée dans Bordewich *et al.* (2009), elle est implicitement utilisée dans les preuves des théorèmes. La présence d'une telle séquence d'opérations SPR est difficile à prouver théoriquement dans le cas de LS et BD, mais les résultats d'une simulation que nous avons conduite à cet effet, suggèrent qu'elle doit aussi exister dans le cas des deux dernières mesures.

3.2.4 Validation des transferts horizontaux de gènes

L'analyse par bootstrap est utilisée pour estimer un intervalle de confiance associé aux arêtes internes des arbres phylogénétiques (Felsenstein, 1985). Ici nous étendons la procédure de validation des transferts horizontaux, initialement proposée dans Makarenkov *et al.* (2006), pour estimer le support de bootstrap des transferts horizontaux de gènes inférés. Les trois stratégies suivantes peuvent être adoptées pour évaluer la fiabilité des THG obtenus :

Première stratégie : les séquences utilisées pour construire les arbres d'espèces et de gène sont répliquées.

Les arbres d'espèces et de gène sont inférés à partir des séquences répliquées avec les mêmes méthodes utilisées pour reconstruire les arbres originaux d'espèces et de gène. Pour tous les THG appartenant au scénario original, nous vérifions s'ils apparaissent dans le scénario généré avec les arbres inférés à partir des réplicats. Cette vérification est effectuée en comparant les opérations SPR correspondantes. Dans cette étude, deux THG (ou opérations SPR) sont considérés comme équivalents si et seulement si les bipartitions des deux arêtes donneuse (e.g., l'arête (x,y) sur la figure 3.3) et receveuse (e.g., l'arête (z,w) sur la figure 3.3) sont équivalentes dans les deux transferts (i.e., les topologies des sous-arbres du donneur et du receveur peuvent être différentes, mais les espèces sont les mêmes pour les deux transferts comparés). Une autre solution, plus stricte, serait de considérer que ces deux transferts sont identiques si et seulement si les sous-arbres donneur et receveur sont identiques dans les deux transferts (i.e., la distance RF entre eux est égale à 0). Comme les données considérées sont des réplicats, une telle stratégie produit habituellement de faibles scores de bootstrap, particulièrement, pour des phylogénies mal résolues. Il est à noter que les ensembles de jeux de données répliquées ne donnent pas toujours lieu à des arbres d'espèces et de gène dont l'arête de la racine est exactement la même que dans les arbres originaux. Si une arête induisant une bipartition identique à l'arête racine des arbres de référence n'existe pas dans l'arbre pseudo-répliqué, alors la racine peut être placée sur une arête produisant la bipartition la plus proche, en termes de la distance de Hamming, de la bipartition induite par l'arête de la racine de l'arbre original. Une telle stratégie de positionnement de la racine est utilisée pour réduire le nombre de THG détectés avec les données répliquées (une stratégie alternative serait de faire l'enracinement avec un *outgroup* ou d'utiliser l'arête médiane de l'arbre).

Deuxième stratégie : seules les données de séquences utilisées pour construire l'arbre de gène sont répliquées.

Les séquences utilisées pour construire l'arbre d'espèces ne sont pas ré-échantillonnées. L'arbre d'espèces est alors fixé et gardé constant. Dans ce cas, nous devons vérifier qu'il est hautement fiable (e.g., il a des hauts scores de bootstrap). Par exemple, l'arbre d'espèces peut être inféré en utilisant les informations taxonomiques appropriées disponibles sur le site du NCBI (NCBI Handbook, 2002) ou celui de "L'arbre de la vie" (Maddison et Schultz, 2004). La situation où la bipartition correspondant à l'arête de la racine dans l'arbre de gène original n'est pas retrouvée dans l'arbre inféré à partir des réplicats, peut être traitée comme dans le cas précédent. Cette stratégie donne habituellement des scores de bootstrap des transferts plus élevés que la première stratégie.

Troisième stratégie : le bootstrap des transferts peut être calculé entre deux topologies d'arbres.

Contrairement au bootstrap traditionnel qui nécessite des séquences pour calculer les scores de bootstrap, le bootstrap de THG peut être effectué même si les séquences permettant l'inférence des topologies des arbres d'espèces et de gène ne sont pas disponibles. Précisément, nous pouvons tout d'abord exécuter notre programme avec une option de recherche exhaustive, produisant la liste de tous les scénarios de transferts de coût minimal; cette option est aussi disponible dans le programme *LatTrans* (Hallett et Lagergren, 2001) qui a une complexité algorithmique exponentielle en fonction du nombre de THG. Dans notre stratégie, cette option consiste à vérifier toutes les opérations SPR satisfaisant la contrainte de sous-arbres qui minimisent à chaque étape la valeur du critère d'optimisation sélectionné. Dès que cette liste est établie, nous pouvons calculer les scores de bootstrap de THG en estimant le taux d'occurrences de chaque THG dans cette liste.

Quand les séquences servant à la reconstruction des arbres d'espèces et de gène sont disponibles, les combinaisons des stratégies (1 et 3) ou (2 et 3) peuvent être aussi considérées pour évaluer le support de bootstrap. Dans un cas général, les Formules 2 et 3 peuvent être utilisées pour calculer le score de bootstrap *HGT_BS* du transfert t :

$$HGT\_BS(t) = \left(\sum_{1 \le i \le N_T} \sum_{1 \le j \le N_{T'}} \left(\sum_{1 \le k \le N_{ij}} \frac{\sigma_{k,ij}(t)}{N_{ij}} \times 100\% \right) \right) / (N_T \times N_{T'}), \text{ et} \quad (2)$$

$$\sigma_{k,ij}(t) = \begin{cases} 1, \text{ si } t \text{ est dans le scénario de coût minimal } k \text{ pour les arbres d'espèces } T_i \text{ et de gène } T_j', \\ 0, \text{ sinon} \end{cases} \quad (3)$$

où NT et NT' sont, respectivement, le nombre d'arbres d'espèces et de gène générés à partir des réplicats et N_{ij} est le nombre de scénarios de coût minimal obtenus quand l'algorithme est appliqué à l'arbre d'espèces T_i et l'arbre de gène T_j'. Le score de bootstrap d'un scénario peut être défini comme le produit de tous les scores de bootstrap individuels faisant partie du scénario obtenu. Une comparaison de la technique de validation proposée avec la méthode de validation incluse dans le logiciel *PhyloNet* (Than *et al.*, 2008a) est présentée dans la section suivante.

3.3 Simulations Monte-Carlo

3.3.1 Description des simulations

Une étude Monté-Carlo a été conduite pour tester l'habilité du nouvel algorithme à retrouver des transferts corrects. Deux types de simulations ont été conduits : dans la première, nous avons considéré les arbres de gènes avec différents niveaux de confiance (leur bootstrap moyen se situe entre 60 et 100%). Nous avons alors mesuré et comparé le taux de détection pour les quatre critères d'optimisation de notre algorithme, ainsi que pour la méthode *LatTrans*. Dans le second, nous avons assumé que les arbres de gènes ne contiennent pas d'incertitudes et les simulations sont effectuées sans considérer les séquences (les arbres d'espèces sont supposés être connus dans les deux types de simulations). Nous avons alors examiné les performances du nouvel algorithme en fonction du critère d'optimisation choisi (incluant LS, RF, QD et BD), du nombre d'espèces observées et du nombre de THG. Par la suite, une comparaison détaillée avec les algorithmes *LatTrans* (Hallett et Lagergren, 2001) et *RIATA-HGT* (Nakhleh *et al.*, 2005; Than et Nakhleh, 2008) a été effectuée en utilisant la dissimilarité de bipartitions (BD, Makarenkov et al., 2007; Boc et al., 2010a) comme critère d'optimisation dans notre algorithme. Ce critère a donné les meilleurs résultats parmi les

quatre critères testés. La procédure de simulations inclut les quatre étapes de base décrites ci-dessous.

Première étape : génération d'un arbre d'espèces aléatoire.

Un arbre d'espèces binaire T a été généré en utilisant une procédure de génération d'arbres aléatoires proposée par Kuhner et Felsenstein (1994). La longueur des arêtes de T a été calculée en utilisant une distribution exponentielle. Suivant l'approche de Guindon et Gascuel (2002), nous avons ajouté du bruit aux arêtes de la phylogénie d'espèces pour créer une déviation par rapport à l'hypothèse de l'horloge moléculaire. Toutes les longueurs des arêtes de T ont été multipliées par $1+ax$, où la variable x a été tirée d'une distribution exponentielle ($P(x>k) = \exp(-k)$), et la constante a était le facteur de réglage de l'intensité de la variation. Comme dans Guindon et Gascuel (2002), la valeur de a a été fixée à 0,8. Les arbres aléatoires générés par cette procédure avaient une profondeur de $O(\log(n))$, où n était le nombre d'espèces (i.e., nombre de feuilles dans un arbre phylogénétique binaire).

Deuxième étape : génération des séquences le long de l'arbre d'espèces aléatoire.

Pour le premier type de simulations uniquement, où les arbres de gènes étaient supposés inclure des incertitudes, nous avons utilisé le programme *SeqGen* (Rambaut et Grassly, 1996) pour générer les séquences d'ADN le long des arêtes de l'arbre d'espèces construit à la première étape. Comme *SeqGen* ne donne que les séquences associées aux feuilles de l'arbre, nous avons légèrement modifié son code pour pouvoir extraire les séquences générées qui sont associées aux nœuds internes de l'arbre (en fait, *SeqGen* génère ces séquences ancestraux, mais ne les affiche pas). Le programme *SeqGen* a été utilisé avec le modèle de substitution de nucléotides HKY (Hasegawa *et al.*, 1985), le modèle de taux d'hétérogénéités assignant différents taux aux différents sites selon une distribution *Gamma* (avec le paramètre *shape* égale à 1.0) et le rapport (TS/TV) égal à 2.0. Ces configurations ont été sélectionnées dans le but de rendre les paramètres de simulations similaires à ceux utilisés dans la section d'exemples. Les séquences d'ADN avec 100, 500, 1000, 5000 et 10000 nucléotides ont été générées.

Troisième étape : simulation des transferts horizontaux.

Pour chaque arbre d'espèces *T*, nous avons généré les arbres de gènes avec le même nombre de feuilles en appliquant un nombre fixe de mouvements SPR aléatoires (représentant les THG) de ses sous-arbres. Un modèle satisfaisant les contraintes d'évolutions (figure 3.1) a été implémenté pour générer les THG aléatoires. Pour chaque arbre d'espèces, les arbres de gènes englobant un nombre différent de THG, variant de 1 à 10, ont été générés. Dans le premier type de simulations où les séquences ont été analysées, nous avons procédé comme suit : après chaque opération SPR, nous avons régénéré, en utilisant *SeqGen*, les séquences associées à tous les nœuds des sous-arbres déplacés. Cette régénération a été initiée à partir de la séquence racine de tous les sous-arbres déplacés. La séquence racine a été initialisée avec celle associée au nœud interne, le plus proche de la racine, de l'arête receveuse. Pour chaque taille de séquences, différents taux de substitutions ont été simulés. Différentes longueurs d'arêtes ont été considérées dans le but d'atteindre la variation de taux de substitutions (voir Posada et Crandall, 2001 pour plus de détails). Ces variations ont mené à des arbres de gènes avec différents scores de bootstrap, variant de 60 à 100%. Par exemple, pour les arbres de gènes avec 50 feuilles et des séquences d'ADN de 1000 nucléotides, la moyenne de longueurs d'arêtes de 4,3 était nécessaire pour obtenir un score de bootstrap de 100%, alors qu'un score moyen de 60% correspondait à une longueur moyenne d'arêtes beaucoup plus petite, et égale à 0,08. Pour obtenir une longueur moyenne d'arêtes d'un arbre de gène nécessaire, nous avons divisé toutes les longueurs d'arêtes de l'arbre d'espèces correspondant par une valeur constante prédéfinie, que nous avons calculée à l'étape 1. En utilisant le programme *Seqboot* du paquet *PHYLIP* (Felsenstein, 1989), nous avons créé 100 réplicats de chaque ensemble de données générées. Les arbres de maximum de vraisemblance (ML) ont été inférés à partir des séquences originales et répliquées en utilisant la méthode *PHYML* (Guindon et Gascuel, 2003). Tous les paramètres de *PHYML* utilisés étaient identiques à ceux employés dans *SeqGen*. Toutes les phylogénies inférées à partir des séquences d'ADN ont été classifiées dans une des 4 catégories selon le score moyen de bootstrap (intervalles : 60 à 70%, 70 à 80%, 80 à 90% et de 90 à 100% ; le programme *PHYML* permet aussi de calculer les scores de bootstrap). Les arbres de gènes dont les scores de bootstrap moyens étaient inférieurs à 60% ont été exclus de l'analyse. Une distribution uniforme des arbres de gènes dans chacun des 4 intervalles a été maintenue.

Simulations : les algorithmes comparés.

Les résultats illustrés dans les figures 3.6, 3.7, 3.8, 3.9 et 3.10 ont été obtenus dans les simulations effectuées avec des arbres phylogénétiques aléatoires avec 10, 20, ..., 100 feuilles. Pour chaque taille d'arbres, nombre de THG, taille des séquences et taux de substitutions (les deux derniers paramètres ont été considérés uniquement dans les simulations avec des séquences), 1000 jeux de données répliqués ont été générés (à l'exception de *LatTrans* dans le cas de la figure 3.7). Dans les simulations avec les arbres et les séquences, les quatre stratégies basées sur LS, RF, QD et BD, et l'algorithme de recherche exhaustive *LatTrans* ont été comparés (voir les figures 3.7, 3.8, 3.9, 3.10). Puis, la stratégie basée sur BD a été comparée à l'algorithme *RIATA-HGT* (dans ce cas, la comparaison a été aussi conduite pour les arbres non-binaires).

3.3.2 Résultats des simulations

3.3.2.1 Comparaison de notre algorithme utilisant LS, RF, QD et BD comme critère d'optimisation et de *LatTrans*

Nous avons premièrement comparé les quatre stratégies algorithmiques discutées dans des simulations avec des données de séquences. Le comportement du taux de détection des THG par rapport au nombre de THG est présenté à la figure 3.6. Les résultats des algorithmes utilisant LS, RF, QD et BD, et ceux utilisant l'algorithme *LatTrans*, sont présentés séparément pour chacun des intervalles de confiance des arbres de gènes sélectionnés (i.e., 60 à 70%, 70 à 80%, 80 à 90% et 90 à 100%). Le taux de détection (i.e., vrais positifs) a été mesuré comme un pourcentage de recouvrement des transferts présents dans le scénario original. L'algorithme basé sur BD a été en général plus efficace que les stratégies utilisant LS, RF et QD et que *LatTrans* en termes de taux de détection (figure 3.6). Ses performances sont plus notables pour les arbres de gènes avec un haut niveau de confiance, allant de 80 à 100% (figure 3.6c-d), quand il est comparé aux stratégies utilisant LS, RF et QD, et pour les arbres de gènes avec un faible niveau de confiance, allant de 60 à 80% (Figure 3.6a-b), quand il est comparé à *LatTrans*.

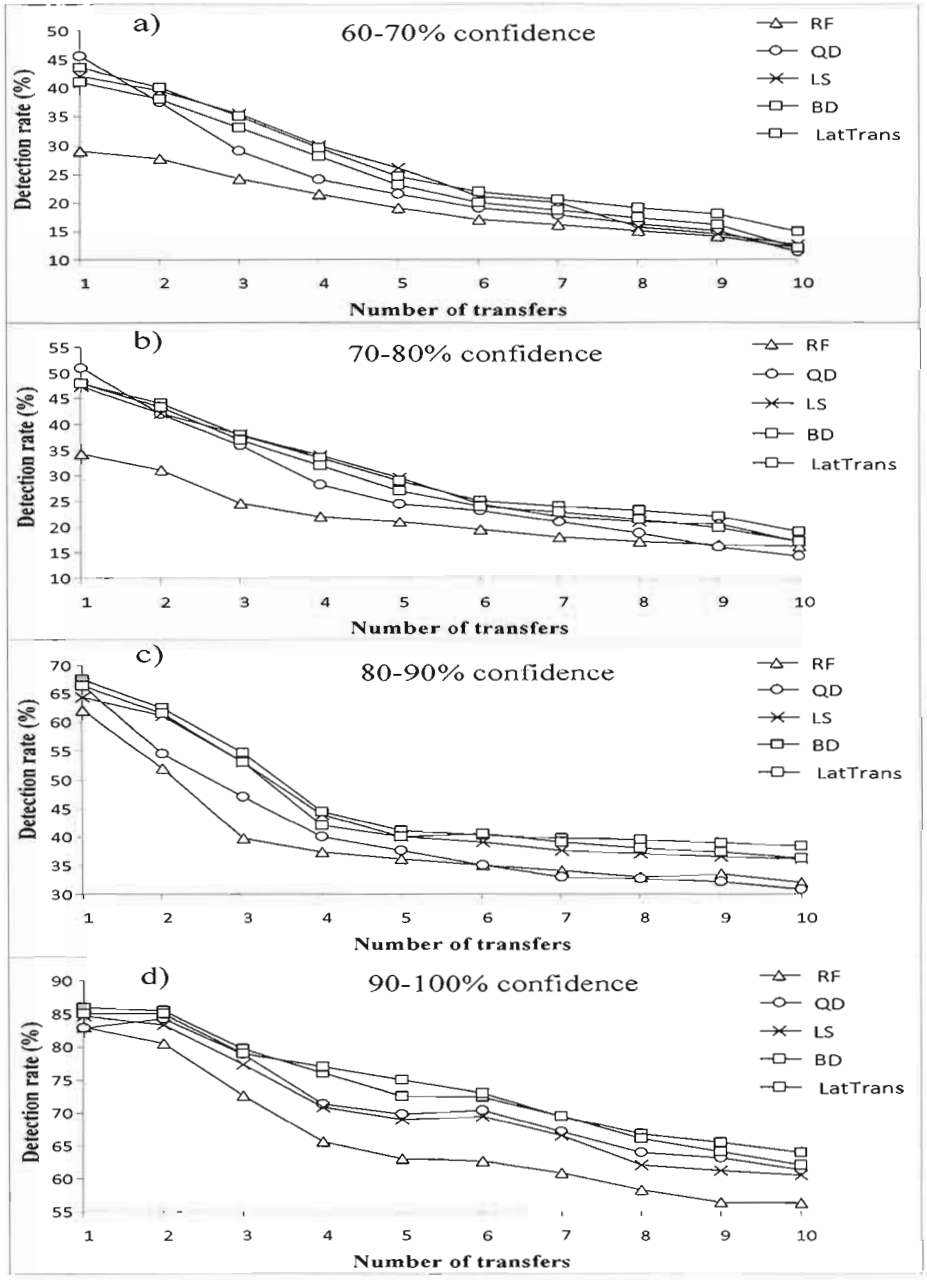


Figure 3.6 Taux de détection des transferts horizontaux en fonction du nombre de THG pour quatre niveaux de confiance des arbres de gène. Chaque graphique est associé à un intervalle de confiance associé à l'arbre de gène : (a) de 60 à 70%, (b) de 70 à 80%, (c) de 80 à 90%, et (d) de 90 à 100%. Chaque point du graphique représente le résultat moyen obtenu pour des arbres aléatoires de 10, 20 ... 100 feuilles et des séquences d'ADN de 100, 500, 1000, 5,000 et 10,000 nucléotides ; 1000 réplicats ont été générés pour chaque combinaison de paramètres pour les stratégies algorithmiques basées sur LS, RF, QD et BD, et 100 réplicats pour *LatTrans*.

Il est intéressant de noter que les algorithmes utilisant BD et LS aussi bien que *LatTrans* ont produit des résultats très similaires quand le nombre de THG était faible.

Pour les arbres de gènes avec un haut niveau de confiance (figure 3.6d), le critère BD et *LatTrans* ont produit des résultats très stables qui étaient meilleurs que ceux fournis par QD et LS. Cependant, pour les arbres de gènes avec un support de bootstrap moyen ou faible (figure 3.6a-c) l'utilisation du critère LS permet de surclasser les critères QD et RF, et mène souvent aux résultats très proches de ceux générés par BD et *LatTrans*. Sans surprise, notre algorithme utilisant le critère RF a été habituellement le pire parmi les cinq techniques comparées quel que soit le niveau de confiance de l'arbre de gène et le nombre de THG.

3.3.2.2 Résultats des simulations avec des arbres sans incertitude

Nous avons étudié le comportement des algorithmes basés sur LS, RF, QD et BD sous la condition d'exactitude de l'arbre de gène. La figure 3.7a montre les taux de détection moyens correspondant aux quatre critères considérés. La figure 3.7b montre la précision des quatre stratégies en termes de recouvrement des scénarios complets (l'ensemble des transferts simulés) de THG : Dans les deux cas (figure 3.7a-b), la stratégie basée sur BD surclasse clairement les stratégies basées sur RF, LS et QD.

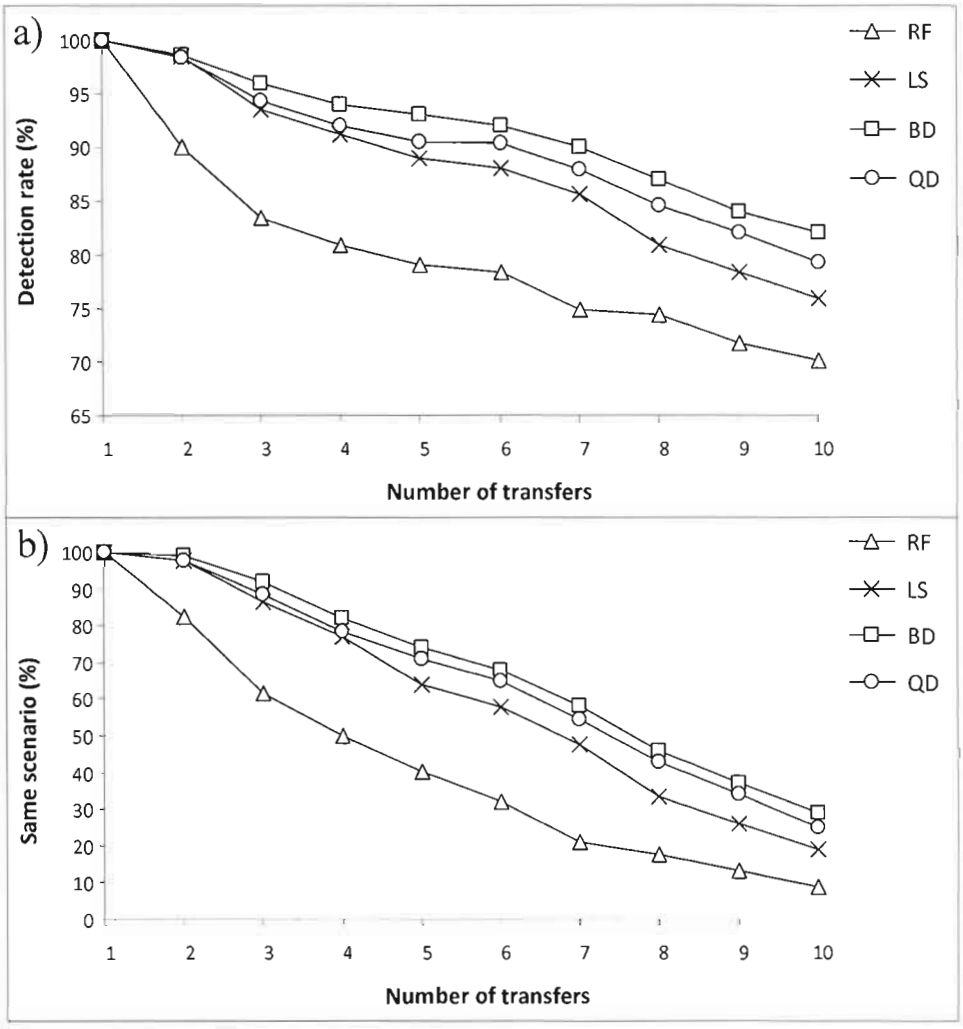


Figure 3.7 Comparaison des stratégies algorithmiques utilisant différents critères d'optimisation (RF, LS, BD et QD). Le taux de détection des transferts corrects (a), et le taux de scénarios de THG complets retrouvés (b) sont représentés en fonction du nombre de transferts, variant de 1 à 10.

Les résultats obtenus avec QD s'améliorent quand le nombre de THG diminue (figure 3.7a), et ils sont seulement légèrement inférieurs à ceux obtenus avec BD lorsqu'on identifie les scénarios complets de THG (figure 3.7b). La distance RF est la pire des quatre stratégies en termes de taux de détection des THG et d'identification du nombre total de transferts. Les performances de la stratégie basée sur BD sont les plus intéressantes en termes de taux de détection des THG.

3.3.2.3 Comparaison détaillée avec *LatTrans*

La stratégie algorithmique basée sur BD a été comparée à l'algorithme *LatTrans* dans le cas où l'on ne dispose que des topologies d'arbres. La comparaison de ces algorithmes basés sur les distances a été conduite en considérant le taux de détection des THG et le temps d'exécution. Notons que la complexité temporelle de l'algorithme *LatTrans* est $O(2\tau n^2)$, où τ est le nombre de transferts et n est le nombre de feuilles dans l'arbre (Hallett et Lagergren, 2001). La figure 3.8a-f représente le taux de détection des deux algorithmes en fonction du nombre de feuilles de l'arbre et du nombre de transferts générés. Les diagrammes de la figure 3.8a-b montrent le taux de détection des transferts en fonction du nombre de feuilles et du nombre de THG générés. Comme *LatTrans* doit produire comme solution une liste de tous les scénarios de coût minimal, nous avons toujours sélectionné le premier scénario de la liste pour calculer les taux de transferts (selon Beiko et Hamilton, 2006, *LatTrans* peut cependant manquer certains scénarios de coût minimal dans des larges phylogénies). Sans surprise, le taux de détection augmente avec l'augmentation du nombre de feuilles. En observant le taux de détection par rapport au nombre de feuilles, *LatTrans* surclasse légèrement l'algorithme basé sur BD (figure 3.8a) pour les arbres de 50 à 70 feuilles, tandis que notre algorithme est meilleur dans tous les autres cas. En observant le taux de détection par rapport au nombre de THG (figure 3.8b), notre algorithme est meilleur pour les grands nombres de THG (5 à 10) et plus faible pour les petits nombres (1 à 3). La figure 3.8c-d montre la précision des deux algorithmes quand nous ne considérons que le nombre de THG retrouvés. Quand le nombre total de THG est retrouvé correctement, la seule possibilité pour ne pas détecter la position ou la direction exacte de certains THG demeure l'existence de plusieurs scénarios de coût minimal ou quasi-minimal (si un scénario de coût quasi-minimal est trouvé). Par exemple, un transfert de direction opposée menant à la même solution (i.e., le même arbre de gène) induit une variante d'un scénario de coût identique (voir Maddison, 1997 pour plus de détails sur les transferts opposés, et Addario-Berry *et al.* (2003) pour une discussion sur les scénarios de coût minimal et quasi-minimal). Il est important de noter que parfois *LatTrans* génère des scénarios qui ne satisfont pas les contraintes d'évolution (e.g., dans certains cas des scénarios cycliques, voir la figure 3.1 b et d, sont trouvés par cette méthode). En moyenne, l'algorithme basé sur BD et *LatTrans* étaient capables de prédire correctement le nombre total de THG dans 91,1% et 92,5% des cas, respectivement (figure 3.8c-d). Nous avons aussi mesuré le

pourcentage d'instances quand les algorithmes comparés étaient capables de trouver un scénario complet (figure 3.8e-f). Un scénario complet de THG est identifié si tous les transferts trouvés par un algorithme sont présents dans le scénario original et si leur nombre est aussi correct.

Généralement, l'algorithme basé sur BD a surclassé *LatTrans* en termes de recouvrement de scénarios complets. Cet avantage a principalement été du à la présence de transferts violant les contraintes d'évolution discutées (figure 3.1) dans certains scénarios générés par *LatTrans*. La complexité polynomiale de notre algorithme et l'amélioration de ses résultats, en comparaison avec *LatTrans*, quand le nombre de feuilles ou de THG augmentent, le rend particulièrement intéressant pour l'analyse de larges phylogénies englobant plusieurs conflits topologiques dus aux transferts horizontaux de gènes.

Finalement, nous avons aussi comparé le temps d'exécution des deux algorithmes. Comme précédemment, les performances algorithmiques ont été évaluées en fonction du nombre de THG (figure 3.9a) et du nombre de feuilles (figure 3.9b). Les simulations ont été effectuées sur un PC équipé d'un processeur Pentium IV dual-core de 3.2 GHz et 4 Go de RAM. Les courbes illustrées sur la figure 3.9 confirment qu'à partir de 30 feuilles et 7 transferts, notre algorithme produit un gain très significatif dans le temps d'exécution.

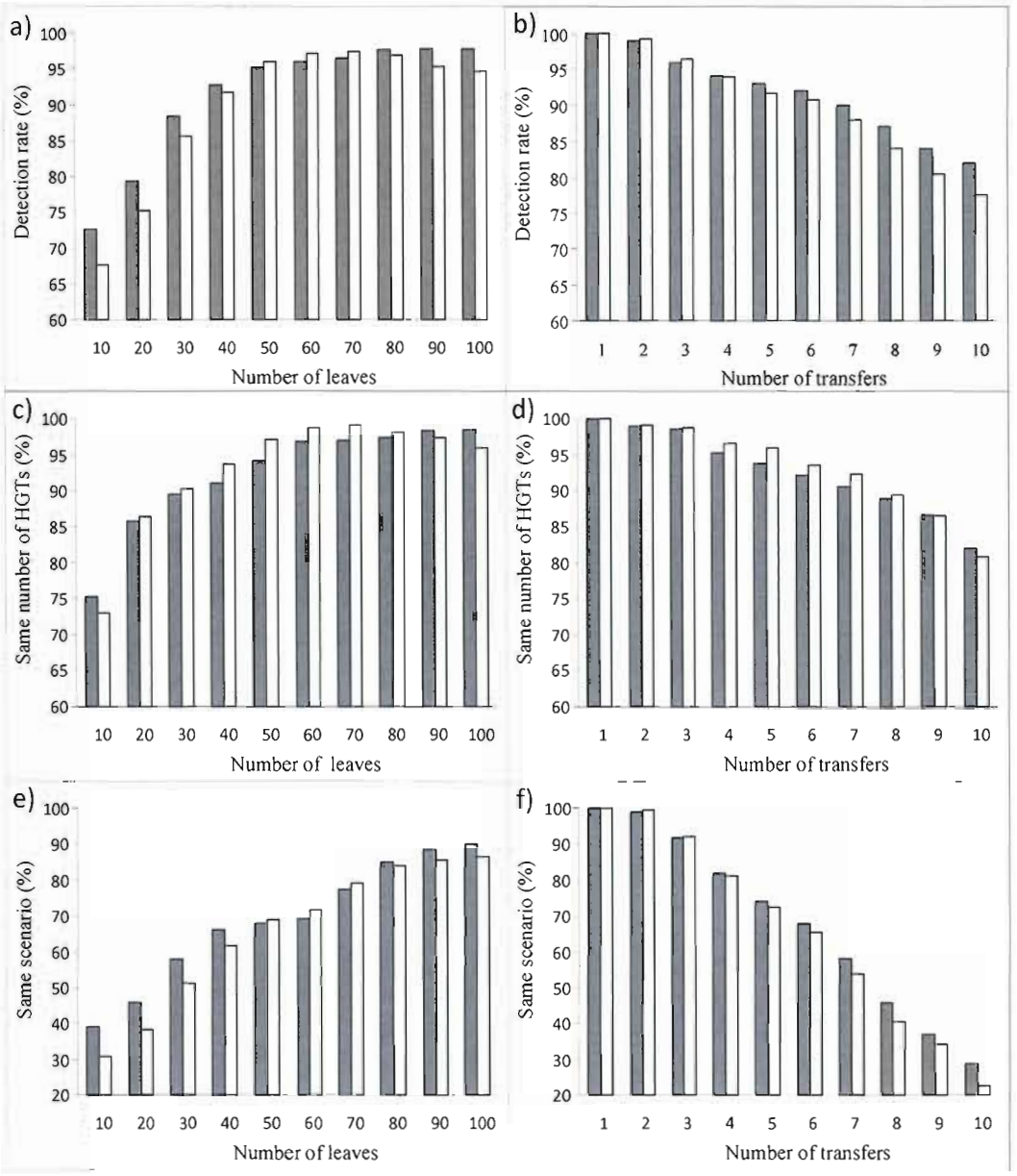


Figure 3.8 Comparaison de la stratégie basée sur BD (▣) avec *LatTrans* (□). Les graphiques (a) et (b) représentent le taux de détection des transferts corrects, (c) et (d) le nombre total correct de THG, et (e) et (f) le taux de détection des scenarios complets de THG.

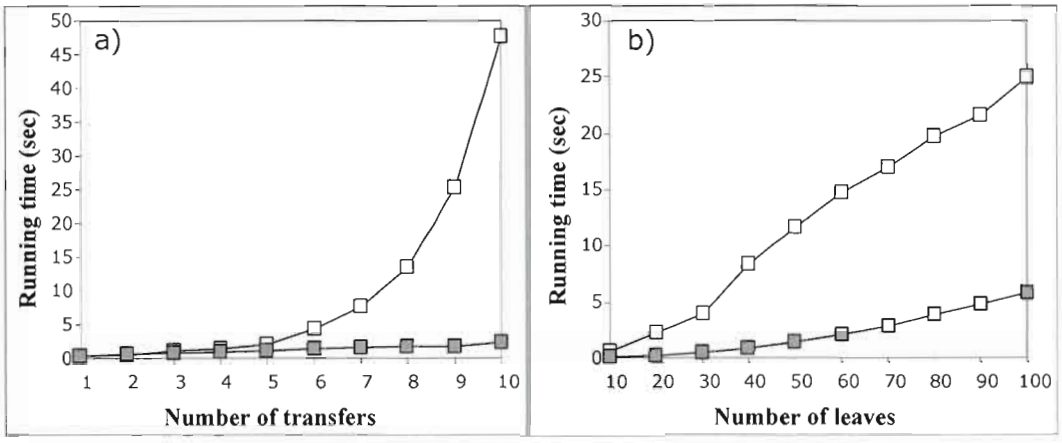


Figure 3.9 Comparaison de *HGT-Detection* (▣) avec *LatTrans* (□) en termes de temps d'exécution. La portion (a) montre le temps d'exécution de l'algorithme en fonction du nombre de transferts et la portion (b), en fonction du nombre de feuilles de l'arbre.

3.3.2.4 Comparaison avec *RIATA-HGT*, *HorizStory* et *EEEP*

À part l'algorithme *LatTrans* (Hallett et Lagergren, 2001), qui est supposé inférer tous les scénarios de THG de coût minimal, mais qui est exponentiel sur le nombre de transferts, plusieurs stratégies heuristiques ont récemment été développées pour détecter les transferts horizontaux de gènes. Parmi les plus populaires, nous mentionnons *HorizStory* (MacLeod *et al.*, 2005), Efficient Evaluation of Edit Paths (ou *EEEP* ; Beiko et Hamilton, 2006) et *RIATA-HGT* (Nakhleh *et al.*, 2005). Tous ces algorithmes visent à détecter les THG en réconciliant une paire de phylogénies, d'espèces et de gène. Le paquetage *PhyloNet* (Than *et al.*, 2008a) inclut une implémentation étendue de l'algorithme *RIATA-HGT* avec plusieurs améliorations algorithmiques pour calculer des solutions multiples et pour manipuler des arbres non-binaires (Than et Nakhleh, 2008). Les résultats des simulations présentées dans Than *et al.* (2007) et Than et Nakhleh (2008) suggèrent que la nouvelle version de *RIATA-HGT* surclasse significativement en termes de rapidité les algorithmes *HorizStory*, *EEEP* et *LatTrans*, et fonctionne au moins aussi bien que *LatTrans* en termes de précision. Une nouvelle caractéristique importante récemment ajoutée au paquet *PhyloNet* est l'estimation du support de bootstrap des transferts retrouvés (Than *et al.*, 2008b). *RIATA-HGT* ne recouvre pas toujours le scénario de coût minimal, mais les résultats expérimentaux montrent de très bonnes performances empiriques sur des données synthétiques et réelles (Nakhleh *et al.*,

2005). *RIATA-HGT* génère un ensemble de scénarios de THG de la même taille et produit un réseau de consensus des solutions obtenues. D'autre part, la simulation conduite par Beiko et Hamilton (2006, voir la Table 1 et la figure 4 dans leur article) pour comparer les performances des algorithmes *HorizStory*, *EEEP* et *LatTrans* confirme que *LatTrans* surclasse *HorizStory* et *EEEP* en termes de précision de détection des THG. Par exemple, pour les arbres avec 5 à 20 feuilles les trois algorithmes montrent un recouvrement presque parfait des THG (90 à 100% de taux de recouvrement), mais pour de larges arbres (30 à 100 feuilles) les performances de *HorizStory* et *EEEP* chutent significativement (la Table 1 dans Beiko et Hamilton, 2006 montre que pour les arbres avec 100 feuilles, le taux moyen de recouvrement pour *HorizStory* est 33,3%, pour *EEEP* est 70% et pour *LatTrans* est 96,7%). Par conséquent, nous avons décidé de comparer les performances de notre algorithme basé sur BD à *RIATA-HGT* (version 1.6), qui a un nombre de caractéristiques communes avec notre algorithme (e.g., manipulation d'arbres non-binaires, estimation du support de bootstrap) et qui est l'heuristique la plus rapide en termes de temps d'exécution. La comparaison avec *RIATA-HGT* a été conduite sur des données d'arbres en termes de taux de détection des THG et de temps d'exécution. La figure 3.10a-d représente les performances de l'algorithme *RIATA-HGT* et de la stratégie basée sur BD en fonction du nombre de feuilles et de transferts générés. Les simulations ont été menées avec des arbres binaires et non-binaires, et les résultats présentés à la figure 3.10 sont des résultats combinés obtenus pour ces deux types d'arbres. Premièrement, les arbres d'espèces et de gènes ont été générés comme décrit ci-dessus. Deuxièmement, pour effectuer les simulations avec les arbres non-binaires, certains nœuds des arbres binaires ont été fusionnés pour obtenir des multifurcations. Le nombre d'opérations de fusion pour chaque arbre d'espèces a été sélectionné aléatoirement et variait de 1 à n-3 pour un arbre d'espèces de n feuilles. Au total, 100 arbres binaires et 100 arbres non-binaires ont été générés pour chaque paires de paramètres (nombre de transferts horizontaux, qui variait de 1 à 10 ; taille de l'arbre qui variait de 10 à 100 feuilles) ; les arbres de gènes étaient toujours binaires. Les arbres de références générés, utilisés dans les simulations peuvent être téléchargés depuis l'adresse URL suivante : http://www.labunix.uqam.ca/~makarenv/Simulation_trees.zip. La figure 3.10 (a et b) représente l'erreur de détection des THG consistant en une différence moyenne absolue entre le nombre total de transferts générés et retrouvés. Seuls les THG *non-triviaux* ont été pris en

compte dans ces simulations (les THG *triviaux*, possibles dans les arbres non-binaires seulement, sont des transferts entre les arêtes adjacentes, ayant en commun un nœud interne de degré plus grand que 3 ; ils sont seulement nécessaires pour transformer un arbre non-binaire en arbre binaire).

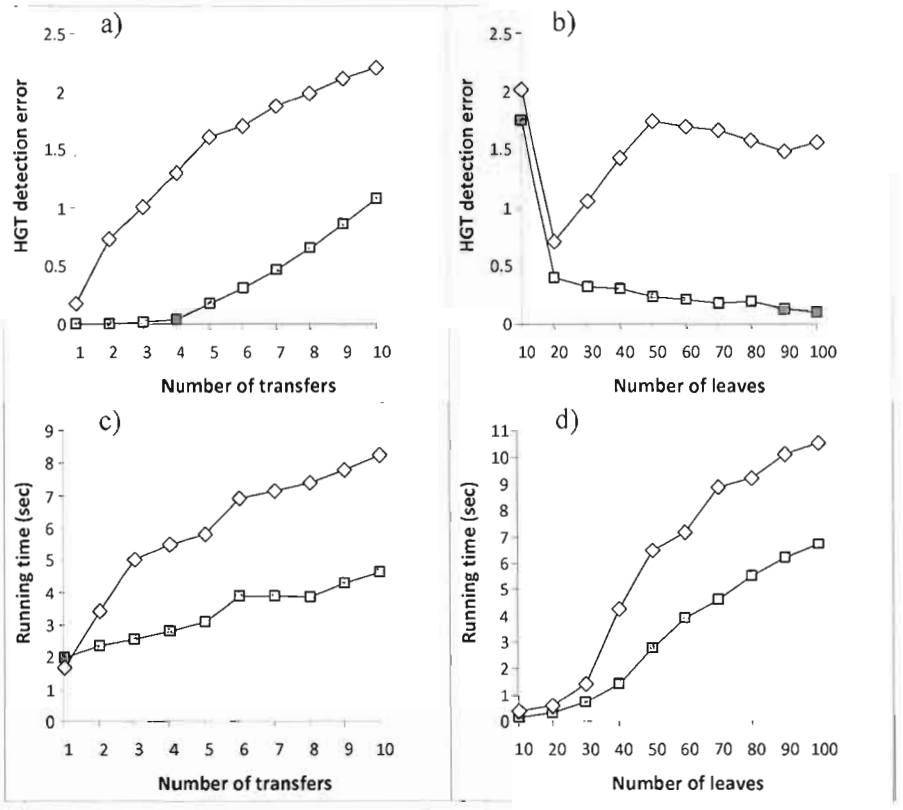


Figure 3.10 Comparaison de *LatTrans* (□) avec *HGT-Detection* (⧈) en termes de taux de détection de transferts et de temps d'exécution. Le taux de détection d'erreurs est représenté en fonction du nombre de transferts (a), et en fonction du nombre de feuilles de l'arbre (b). Le temps d'exécution est représenté en fonction du nombre de transferts (c), et en fonction du nombre de feuilles de l'arbre (d).

La figure 3.10 suggère que l'algorithme basé sur BD surclasse *RIATA-HGT* en termes de taux de détection des transferts horizontaux. Alors que les résultats produits par les deux algorithmes sont très similaires pour les arbres binaires, l'algorithme basé sur BD surpasse clairement *RIATA-HGT* dans le cas des arbres non-binaires. De plus, la précision de notre

algorithme s'améliore quand le nombre de feuilles augmente (figure 3.10b), alors que la précision de *RIATA-HGT* demeure instable (principalement à cause de mauvaises performances dans le cas des arbres non-binaires). En termes de temps d'exécution, l'avantage va aussi à l'algorithme basé sur BD (figure 3.10 c et d). La comparaison des résultats produit par *RIATA-HGT* et par notre algorithme pour les deux jeux de données réelles est faite dans la section Exemples de la présente thèse.

Than *et al.* (2008b) ont aussi proposé une méthode, maintenant incluse dans le paquet *PhyloNet*, pour évaluer le support de bootstrap des THG. La figure 3.11 présente une illustration du calcul de la valeur de support d'une arête de THG par *RIATA-HGT* (voir aussi la figure 8 dans Than *et al.*, 2008b). Dans cette étude, le support de l'arête de THG, $X \rightarrow Y$, ajoutée à l'arbre d'espèces est défini comme le support de bootstrap maximal de toutes les arêtes internes du chemin liant les nœuds Z et X dans l'arbre de gène. Le support de bootstrap de l'événement $X \rightarrow Y$ donné par *RIATA-HGT* est 100% dans ce cas. Ce calcul ne tient pas compte du faible support de bootstrap, de 10%, de l'arête interne séparant les feuilles B et D du reste de l'arbre. Nous pensons que les scores de bootstrap des THG calculés de cette façon sont largement surestimés. En outre, cette façon d'évaluer le support de bootstrap des THG ne prend pas en compte les topologies des phylogénies de gène répliquées (la phylogénie d'espèces étant fixe). Un arbre de gène unique avec les scores de bootstrap donnés de ses arêtes internes n'englobera pas toujours toutes les caractéristiques importantes de l'ensemble des arbres répliqués que nous avons utilisés pour calculer ces scores. Même si le support de bootstrap de chaque clade est indiqué dans un tel arbre unique, l'information clé, concernant le pourcentage d'occurrences quand deux sous-arbres affectés par un THG sont présents ensemble dans les arbres de gène répliqués, est manquante. Dans notre méthode, chaque arbre répliqué est testé, et les statistiques sont combinées (voir les Formules 2 et 3) pour calculer le support de bootstrap des transferts horizontaux. Par exemple, le score de bootstrap du THG $X \rightarrow Y$ (figure 3.11) calculé par notre méthode devrait être 10% au plus.

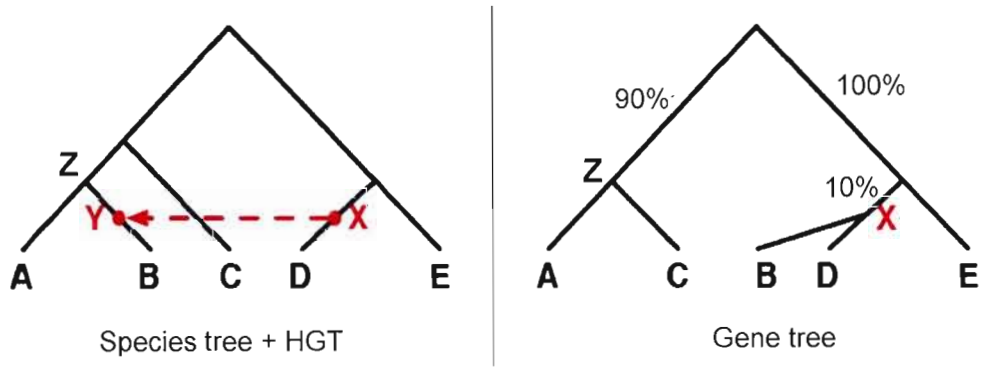


Figure 3.11 Calcul de la valeur de support de bootstrap d'un transfert horizontal par *RIATA-HGT* : le score de bootstrap du transfert $X \rightarrow Y$ est défini comme le score de bootstrap maximum de toutes les branches internes du chemin reliant les nœuds Z et X dans l'arbre de gène (il est égal à 100% dans ce cas). Avec notre méthode, le score de bootstrap du transfert $X \rightarrow Y$ devrait être 10% au plus.

3.4 Exemples

3.4.1 Détection des transferts horizontaux du gène *rpl12e*

Nous avons tout d'abord examiné l'évolution du gène *rpl12e* pour les 14 organismes d'Archées originalement considérés par Matte-Tailliez *et al.* (2002). Les derniers auteurs ont discuté des problèmes rencontrés lors de la reconstruction de certaines parties de la phylogénie des Archées et ont émis une hypothèse selon laquelle des transferts horizontaux ont grandement influencé l'évolution du gène *rpl12e*. Matte-Tailliez *et al.* (2002) ont inféré l'arbre de maximum de vraisemblance (ML) du gène *rpl12e* (figure 3.12) pour 14 organismes d'Archées et l'ont comparé à la phylogénie ML (figure 3.13) basée sur la concaténation de 53 protéines ribosomales (7,175 positions). Le calcul des valeurs du paramètre α et les autres analyses ML, prenant en compte le taux de variation entre les sites et la correction Γ-law pour les 53 protéines concaténées, ont été effectués par Matte-Tailliez *et al.* (2002) en utilisant le programme *PUZZLE* (Strimmer et von Haeseler, 1996). Face à l'incongruence topologique des phylogénies obtenues, les auteurs ont prédit quelques cas de THG du gène *rpl12e*. Plus précisément, le cas du THG entre les clades de Thermoplasmatales (*Ferroplasma*

acidarmanus et *Thermoplasma acidophilum*) et Crenarchaeota (*Aeropyrum pernix*, *Pyrobaculum aerophilum* et *Sulfolobus solfataricus*) a été indiqué comme le plus évident.

Nous avons tout d'abord reconstruit les topologies des arbres d'espèces (figure 3.13) et de gène (figure 3.12) à partir des séquences originales. La détection des THG a été effectuée avec la stratégie algorithmique basée sur la dissimilarité de bipartitions (BD). Cinq transferts nécessaires pour réconcilier les topologies d'espèces et de gène ont été trouvés (ils sont indiqués par des flèches sur la figure 3.13). Le transfert entre le clade (*Halobacterium sp.* et *Haloarcula marismortui*) et l'organisme *Methanobacterium thermoautotrophicum* a été trouvé à la première étape. Son support de bootstrap, calculé en fixant la topologie de l'arbre d'espèces et en répliquant les séquences de l'arbre de gène, est 55%.

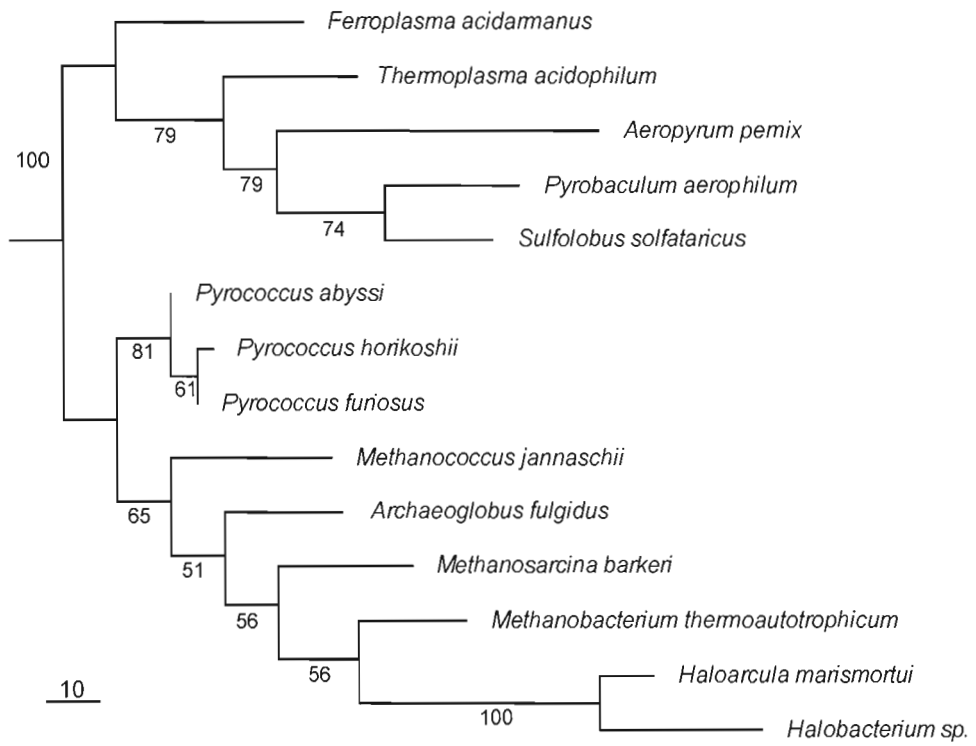


Figure 3.12 Arbre de maximum de vraisemblance du gène *rpl12e*. Les scores de bootstrap associés aux branches de l'arbre, obtenus en utilisant les programmes *Seqboot* et *Protml* (modèle JTT, Jones *et al.*, 1992) du paquetage PHYLIP (Felsenstein, 1989), sont indiqués.

À la deuxième et à la troisième étape, nous avons trouvé les transferts entre *Pyrococcus horikoshii* et *Pyrococcus furiosus* (étape 2), et entre *Sulfolobus solfataricus* et *Pyrobaculum aerophilum* (étape 3). Ces deux transferts lient des espèces proches et ont de faibles scores de bootstrap de 31% et 38%, respectivement. Les faibles scores de bootstrap de ces transferts peuvent être expliqués par la possibilité de THG opposés menant, dans les deux cas, aux mêmes réarrangements topologiques que ceux induits par les transferts obtenus.

Les transferts 4 et 5 lient le clade des Crenarchaeota aux organismes *Thermoplasma acidophilum* et *Ferroplasma acidarmanus*. Les transferts entre ces deux groupes ont été prédits par Matte-Tailliez *et al.* (2002). La direction identique et les scores de bootstrap similaires des THG 4 et 5 suggèrent qu'un unique transfert horizontal, au lieu des deux transferts indiqués, pourrait avoir lieu entre les clades de Thermoplasmatales et Crenarchaeota. Il est à noter que tout algorithme basé sur la minimisation de la distance SPR devrait trouver deux transferts dans ce cas. Un THG unique reliant ces clades serait vraisemblablement caché suite à un artéfact affectant la reconstruction de l'arbre de gène (figure 3.12). Par exemple, si les organismes *Thermoplasma acidophilum* et *Ferroplasma acidarmanus* étaient voisins dans l'arbre de gène, un unique THG du clade des Crenarchaeota au clade des Thermoplasmatales, au lieu des THG 4 et 5 présentés à la figure 3.13, serait suffisant pour obtenir la topologie de l'arbre de gène.

Au total, quatre scénarios de THG de coût minimal ont été trouvés pour les arbres d'espèces et de gène considérés. Tous ces scénarios incluent les THG 1, 4 et 5. Cependant, les transferts 2 et 3 peuvent être présentés comme à la figure 3.13 ou aller dans la direction opposée (voir leur faible score de bootstrap calculé en utilisant les Formule 2 et 3).

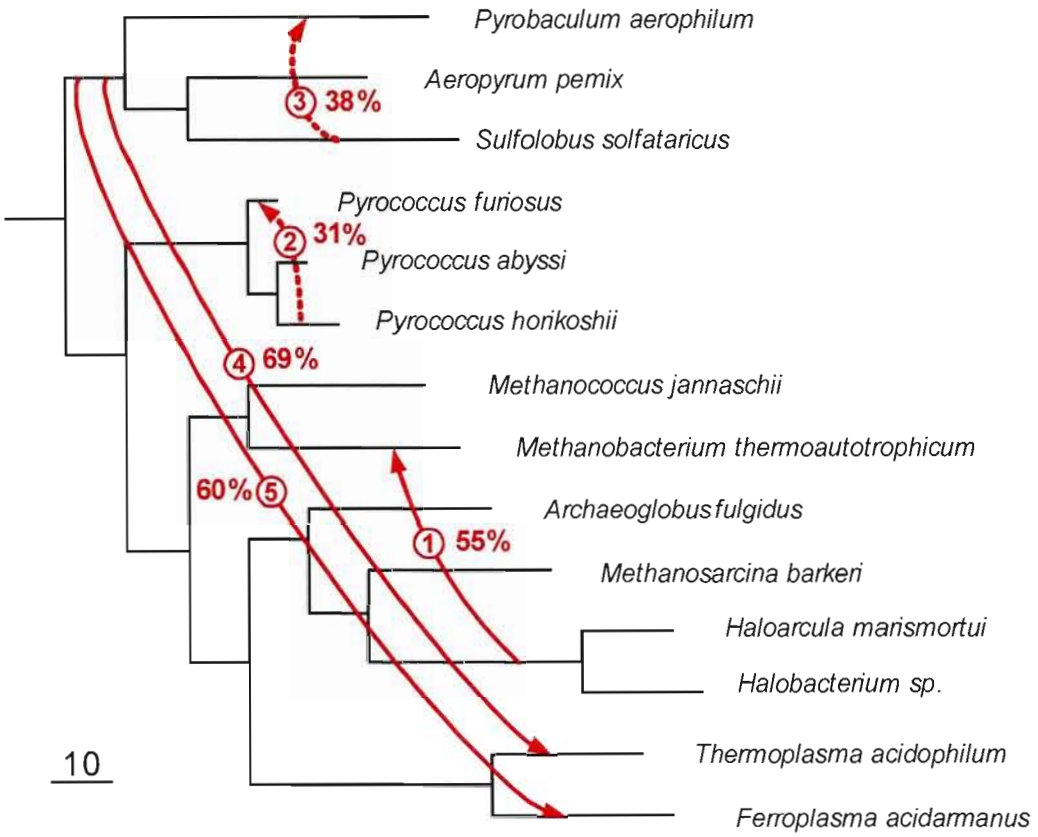


Figure 3.13 Scénario de transferts obtenu par l'algorithme *HGT-Detection* appliqué au jeu de données du gène *rpl12e*. Le score de bootstrap des THG est indiqué à côté de chaque transfert détecté. Les flèches 4 et 5 illustrent les transferts entre les clades des Thermoplasmatales et des Crenarchaeota prédits par Matte-Taillez *et al.* (2002). Les THG avec un bootstrap de 50% ou moins sont illustrés par des flèches en pointillé.

Pour l'exemple des données du gène *rpl12e*, l'algorithme *RIATA-HGT* a trouvé 9 solutions, chacune de taille 5 (figure 3.14). Cinq de ces solutions contredisent la contrainte de la même lignée (ils incluent un THG marqué par *[time violation ?]* dans la sortie du programme, voir l'Annexe B.2) et quatre d'entre elles satisfont toutes les contraintes plausibles d'évolution. La solution représentée sur la figure 3.14 est parmi ces quatre solutions éligibles. Les scores de bootstrap trouvés par *RIATA-HGT* sont indiqués dans la sortie du programme. Ils sont en général plus grands que les scores de bootstrap correspondants calculés par notre méthode, mais nous trouvons ces valeurs largement

surestimées. Par exemple, le score parfait de 100% pour les THG 4 et 5 (figure 3.15) a été trouvé par *RIATA-HGT*, malgré le score de 79% de l'arête liant l'organisme *Thermoplasma acidophilum* et le clade des Crenarchaeota (figure 3.12).

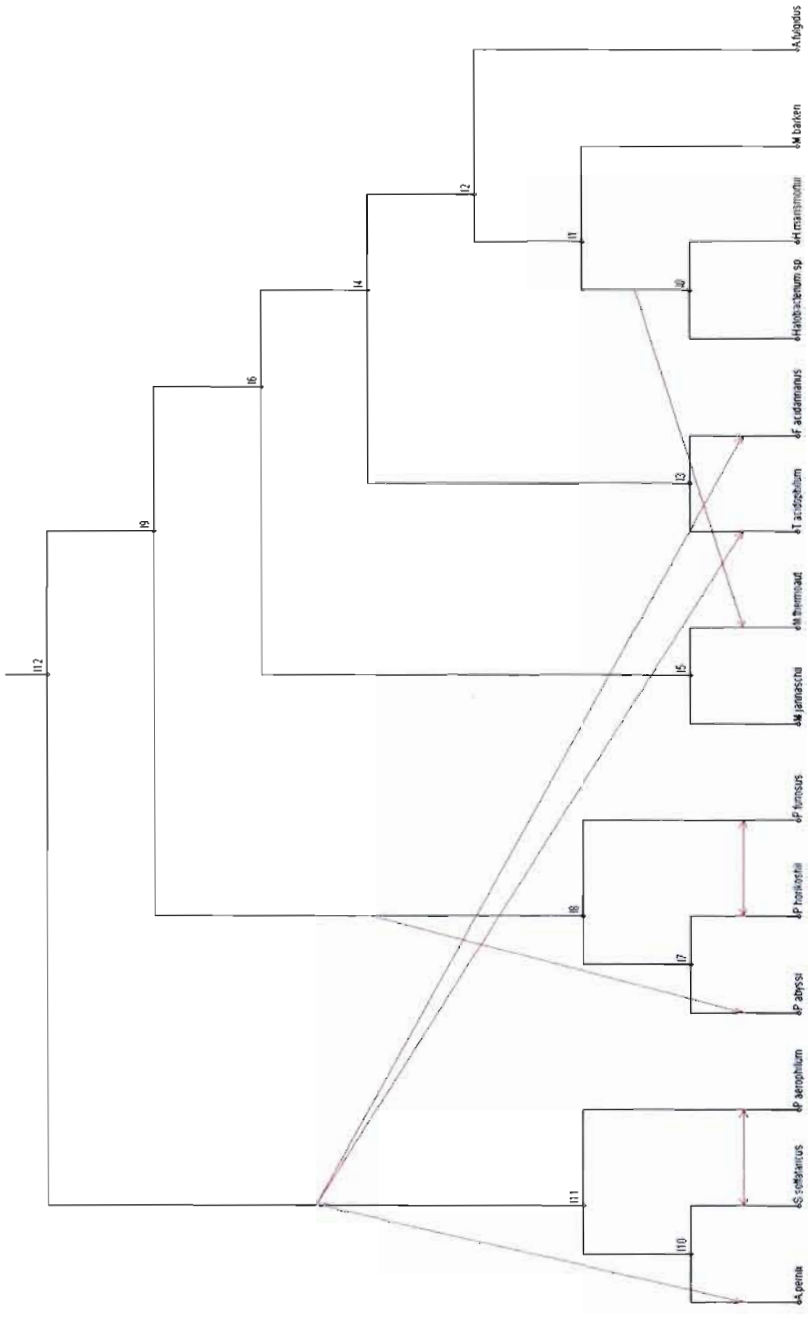


Figure 3.14 Scénarios de transferts obtenus par l'algorithme *RIATA-HGT* appliqué au jeu de données du gène *rpl12e*; 9 solutions, chacune de taille 5 ont été trouvées. Cinq de ces solutions violent la contrainte de la même lignée (figure 3.1a).

3.4.2 Détection des transferts horizontaux de *PheRS* synthétase.

Woese *et al.* (2000) ont analysé, du point de vue évolutif, la relation des aminoacyl-tRNA synthétases (AARS) à leur code génétique. Ils ont trouvé que les AARSs sont très informatifs du point de vue du processus d'évolution. L'analyse de différents arbres phylogénétiques pour un nombre d'AARS considérés a révélé les caractéristiques suivantes : les relations évolutives des AARS sont généralement conformes à la phylogénie d'espèces ; une forte distinction existe entre les AARS de type bactérien et les archées ; le transfert horizontal des gènes AARS entre les bactéries et les archées est asymétrique : les THG d'AARS des archées (groupe de micro-organismes unicellulaires) vers les bactéries est plus prévalent que l'inverse (Boc *et al.*, 2010).

Nous avons examiné l'évolution des séquences du *PheRS* synthétase pour l'ensemble des 32 organismes considérés par Woese *et al.* (2000, figure 2), incluant 24 bactéries, 6 archées et 2 eucaryotes. Comme suggéré par les derniers auteurs, il est pertinent de considérer l'évolution des aminoacyl-tRNA synthétase de haut en bas comme une étude de THG. L'arbre phylogénétique du *PheRS* inféré avec *PHYML* (Guindon et Gascuel, 2003) est montré à la figure 3.15. Le modèle JTT (Jones *et al.*, 1992) a été utilisé et l'arbre a été enraciné entre les bactéries et les archées et eucaryotes en se basant sur la classification taxonomique du NCBI. Woese *et al.* (2000) ont utilisé le même modèle en reconstruisant la phylogénie du gène *PheRS*. Cet arbre est légèrement différent de celui obtenu par Woese *et al.* (2000, figure 2). La plus grosse différence consiste en la présence dans la phylogénie de la figure 3.15 d'un nouveau clade formé par deux eucaryotes (*H. sapiens* et *S. cerevisiae*) et deux archées (*A. fulgidus* et *M. thermoautotrophicum*). Ce clade de quatre espèces, n'apparaissant pas dans l'arbre de consensus (qui n'est pas montré ici), a un faible support de bootstrap et est probablement du à des artéfacts de reconstruction.

Le *PheRS* est la seule synthétase de classe II dans le groupe de codon NUN. Il n'a pas de familles proches à l'intérieure de cette classe. Pour les deux sous-unités α et β du *PheRS*, différentes longueurs se distinguent significativement des sous-unités bactériennes de leurs homologues archées (Woese *et al.*, 2000). Les *PheRS* montrent le schéma classique canonique ; la seule exception étant les spirochètes (i.e., *B. bugdorferi* et *T. pallidum*). Ce

sont des archées qui semblent être reliés à la bactérie *P. horikoshii* dans ce groupe (voir figure 3.15 ou la figure 2 dans Woese *et al.*, 2000). L'analyse de la signature de séquences confirme ce fait.

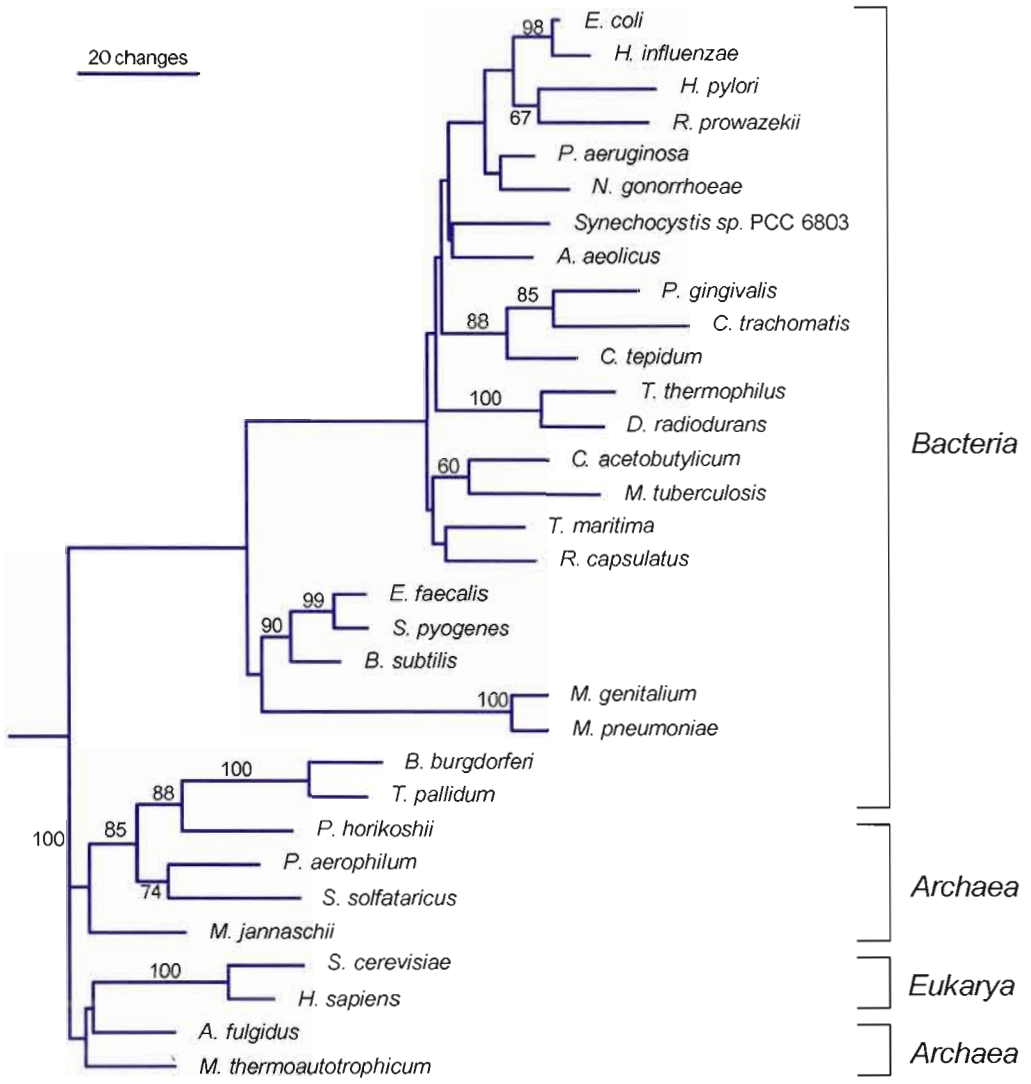


Figure 3.15 Arbre phylogénétique du *PheRS* inféré avec *PHYML*. Les séquences de protéines (171 aa) ont été alignées avec ClustalW (Thompson et al., 1994). Une optimisation de l'alignement a été faite avec MUST (Philippe, 1993). Les régions mal alignées ont été supprimées en utilisant Gblocks (Castresana, 2000); 160 aa ont été

conservés. Les scores de bootstrap supérieurs ou égaux à 60% sont indiqués. L'arbre a été enraciné entre les bactéries et les archées et eucaryotes.

La phylogénie d'espèces correspondant à la classification taxonomique du NCBI (NCBI Handbook, 2002) a aussi été construite (figure 3.16). Notons que dans ce cas la phylogénie d'espèces n'est pas un arbre totalement résolu ; il contient cinq nœuds internes de degré plus grand que trois. Les sept transferts horizontaux non-triviaux (voir la section précédente pour la définition d'un transfert trivial) avec les scores de bootstrap trouvés par notre algorithme sont montrés à la figure 3.16. Au total, l'algorithme a identifié 17 transferts, incluant 10 THG triviaux qui ne sont pas présentés ici. Le transfert numéro 6, ayant le support de bootstrap de 86%, lie l'organisme *P. horikoshii* et le clade des spirochètes, incluant *B. bugdorferi* et *T. pallidum*. Ce score de bootstrap est très proche du plus gros score possible, de 88%, qui peut être obtenu pour ce THG (voir le clade de trois taxons correspondants dans la phylogénie du *PheRS* montrée à la figure 3.16). Ce transfert confirme l'hypothèse que le gène *PheRS* des spirochètes a été affecté par des THG. D'autre part, les faibles scores de bootstrap des trois THG non-triviaux (1, 3 et 5 montrés par des flèches en pointillé sur la figure 3.17) peuvent être expliqués par le faible support de bootstrap des arêtes internes correspondantes dans la phylogénie de gène (figure 3.16). Par exemple, le THG numéro 1 liant l'archée *A. fulgidus* au clade de deux eucaryotes a le score de bootstrap le plus bas (25% seulement). Dans cet exemple, la solution retrouvée en utilisant BD comme critère d'optimisation est présentée. L'utilisation des critères RF, QD et LS, au lieu de BD, mène aux mêmes scénarios de THG qui diffèrent de celui montré à la figure 3.17 seulement par les scores de bootstrap. Pour ces données, un unique scénario de THG de coût minimal avec sept transferts non-triviaux a été trouvé par notre algorithme. Notons que ce jeu de données a été originalement examiné dans Makarenkov *et al.* (2006) en utilisant un algorithme de détection glouton basé sur les critères d'optimisation RF et LS. La solution présentée dans l'article de 2006 (voir la figure 5, page 347), consistait en neuf transferts non-triviaux nécessaires pour transformer l'arbre d'espèces non-binaire de la figure 3.16 en l'arbre de gène binaire de la figure 3.15. Dans cet exemple, l'utilisation du nouvel algorithme a permis d'obtenir un scénario de THG de coût minimal consistant en seulement sept transferts non-triviaux (e.g., le transfert de *H. pylori* à *R. prowazekii* montré sur la figure 5

dans Makarenkov *et al.* 2006 ne fait pas partie du scénario optimal présenté sur la figure 3.16).

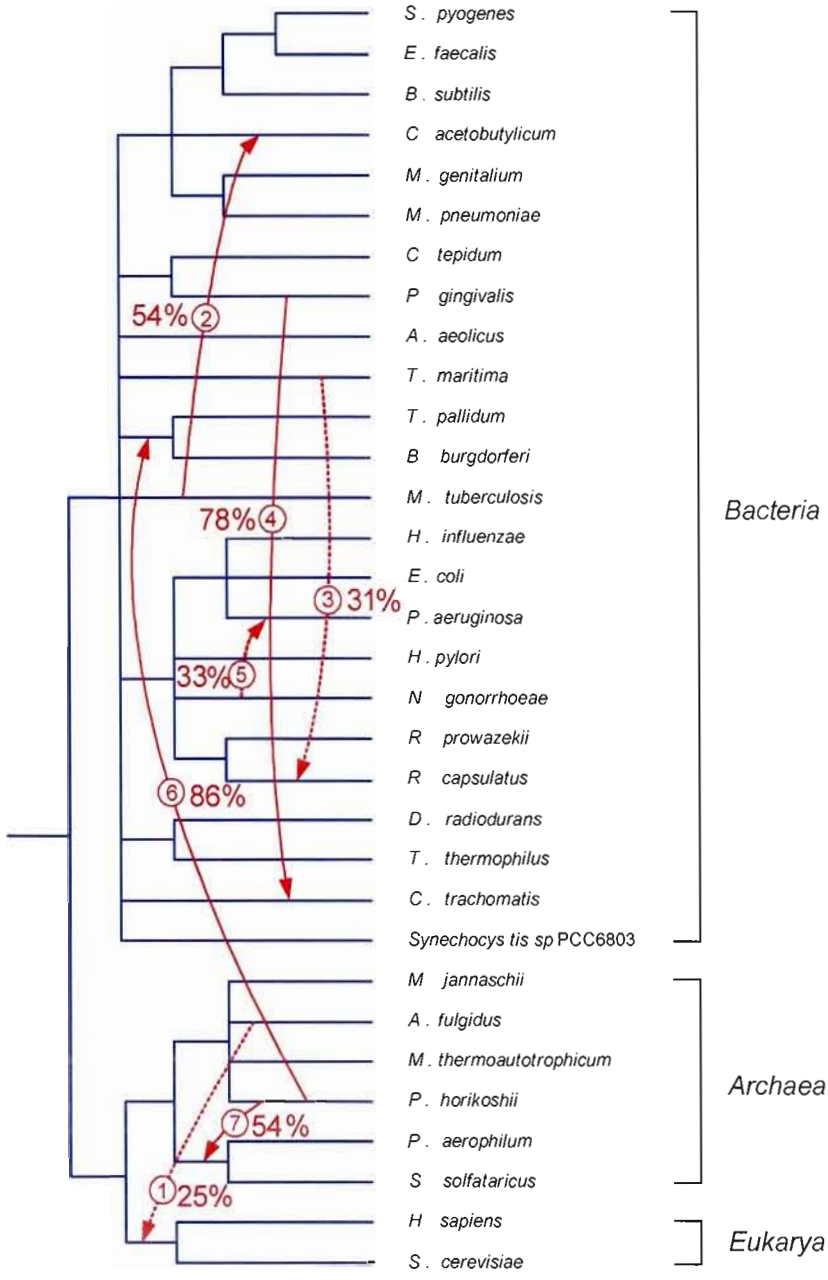


Figure 3.16 Scénario de transferts obtenu par l'algorithme *HGT-Detection* appliqué aux séquences du *PheRS* synthétase. L'arbre a été reconstitué à partir de la

classification taxonomique de NCBI pour les 32 organismes; 7 transferts non-triviaux (indiqués par des flèches), incluant 4 THG avec un score de bootstrap supérieur à 50% et 3 THG inférieur à 50% (flèches en pointillé), ont été trouvés.

Pour ces données, l'algorithme *RIATA-HGT* a généré 12 solutions, chacune de taille 14, incluant seulement des transferts non-triviaux (voir la figure 3.18). Les cinq transformations initiales de l'arbre d'espèces indiquées par les ellipses en pointillé sur la figure 3.17 ont été faites par *RIATA-HGT* avant d'effectuer la détection des THG. Chacune de ces transformations correspond à un transfert trivial. Alors, la solution présentée à la figure 3.17 consiste en 19 THG, comprenant 14 transferts réguliers et 5 transferts triviaux. La solution de coût minimal trouvée par notre algorithme consistait en 7 transferts réguliers et 10 transferts triviaux. Elle n'a pas été trouvée par *RIATA-HGT*.

Comme dans l'exemple précédent, les scores de bootstrap trouvés par *RIATA-HGT* ont généralement été plus élevés que ceux trouvés par notre algorithme (voir la trace d'exécution dans l'Annexe B.3). Par exemple, un score parfait de 100% a été trouvé par *RIATA-HGT* pour le THG allant de l'archaebactérie *P. horikoshii* au clade des spirochètes (THG numéro 6 sur la figure 3.17), alors que le clade regroupant ces organismes dans l'arbre de gène a le support de bootstrap de 88% (figure 3.16).

3.5 Discussion et conclusion

Le transfert horizontal de gène (THG) est un des principaux mécanismes contribuant à la diversification des génomes microbiens. Il est très fréquent chez les différents groupes de gènes des bactéries (Doolittle, 1999). Par exemple, dans une perspective à long terme, il peut être une force dominante, affectant la plupart des gènes chez les procaryotes (Doolittle *et al.*, 2003). En même temps, le THG pose plusieurs risques pour les humains, incluant les gènes résistants aux antibiotiques et se propageant parmi les bactéries pathogènes, l'insertion de nouveaux gènes déclenchant le cancer dans l'ADN humain, de même que les gènes associés à des maladies se propageant et se recombinant pour créer de nouveaux virus et bactéries (Ho, 2002). Dans cette thèse, nous avons décrit un algorithme précis et s'exécutant en un temps polynomial pour l'inférence des transferts horizontaux complets. Chaque THG inséré dans la phylogénie d'espèces aide à réconcilier les topologies des arbres de gène et d'espèces. Les

deux arbres peuvent être inférés à partir de séquences ou de distances et les deux peuvent inclure de l'incertitude. L'algorithme présenté peut se prévaloir d'une optimisation métrique, avec les moindres carrés (LS), ou topologique, avec la distance de Robinson et Foulds (RF), la distance de quartets (QD) ou la dissimilarité de bipartitions (BD), pour prédire les THG. La mesure BD introduite dans cette thèse peut être vue comme un raffinement intéressant de la distance RF. Elle permet de saisir le degré de dissimilarité entre des sous-arbres inégaux, ce que la largement utilisée distance RF ne parvient pas à faire.

La figure 3.18 présente un exemple typique de la situation où la distance RF est inappropriée pour trouver un scénario optimal de transformations SPR. Elle montre un THG dans un arbre binaire avec n feuilles (arbre en forme d'une chenille). Ici, la phylogénie d'espèces T est l'arbre avant le transfert et la phylogénie de gène T' est l'arbre après ce transfert. Donc, la distance SPR entre T et T' est 1, alors que la distance RF entre T et T' est égale à son maximum possible : $2n$-6. Cet exemple suggère que la métrique RF n'est pas vraiment une mesure appropriée pour approximer la distance SPR. D'autre part, la valeur de la dissimilarité de bipartitions entre T et T' est n-3, alors que son maximum pour le cas de deux arbres binaires avec n feuilles est $n(n\text{-}3)/2$ quand n est pair et $(n\text{-}1)(n\text{-}3)/2$ quand n est impair (voir la Proposition 2).

L'algorithme *HGT-Detection* présenté ici a un nombre important de propriétés et d'avantages. Premièrement, les Théorèmes 1 et 2, utilisés dans la procédure algorithmique, lui permet d'inférer les transferts faisant partie d'un (ou de plusieurs) scénario(s) de THG de coût minimal. L'algorithme décrit n'est pas limité à des arbres d'espèces binaires. L'exemple des données de *PheRS* synthétase confirme qu'il peut être utilisé efficacement dans le cas où l'arbre d'espèces n'est pas totalement résolu. Dans ce cas, des THG triviaux seront aussi produits par l'algorithme. Ils doivent être ignorés dans la solution finale. D'autre part, le cas où les arbres de gène et d'espèces ont un nombre différent de feuilles peut être aussi pris en compte. Dans cette situation, nous devons d'abord trouver le sous-arbre maximal d'espèces identiques (i.e., feuilles) présentes dans les deux arbres et supprimer à répétition, dans les deux arbres, toutes les arêtes connectées aux espèces qui ne sont pas incluses dans ce sous-arbre jusqu'à ce que les arbres ne comprennent qu'un ensemble de feuilles identiques. Quand l'opération de suppression est terminée, l'algorithme peut être appliqué comme discuté.

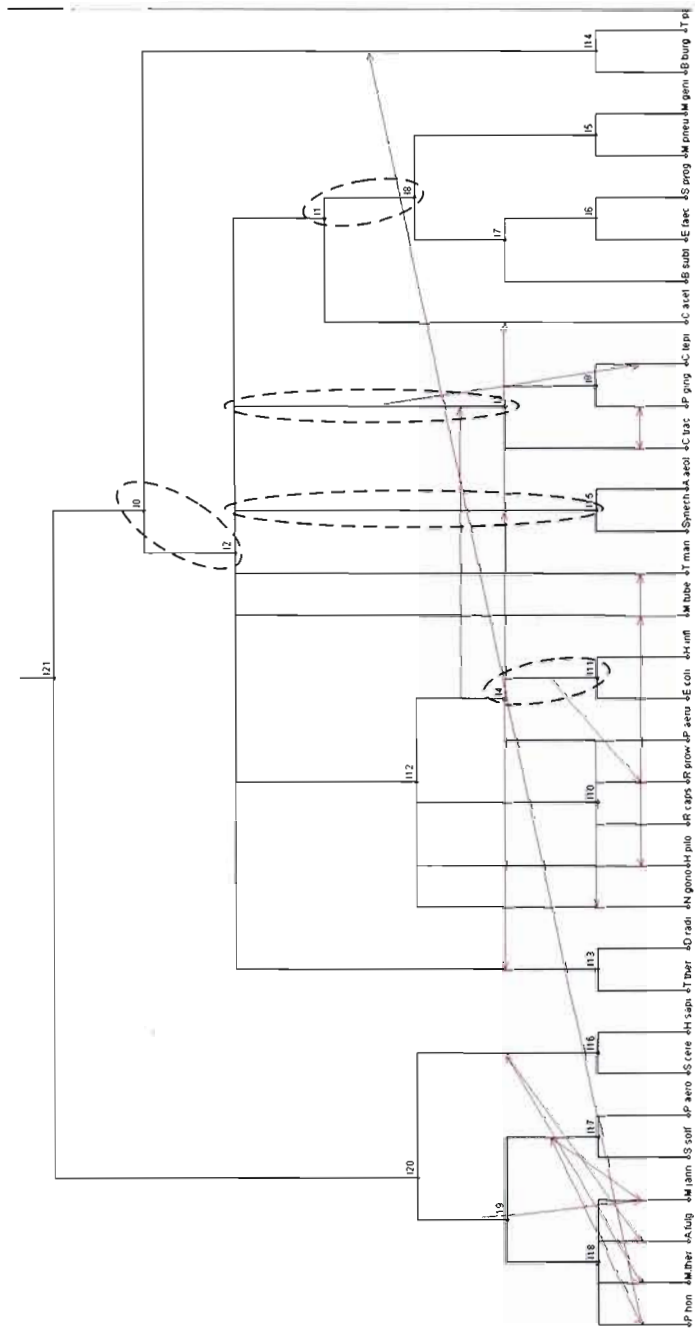


Figure 3.17 Scénarios de transferts obtenus par l'algorithme *RIATA-HGT* appliqué aux séquences du *PheRs* synthétase; 12 solutions ont été trouvées, chacune de taille 14. La solution présentée consiste en 19 THG, dont 14 réguliers et 5 triviaux.

Aussi, la situation où plus d'une copie d'un gène est considérée, peut être prise en compte en introduisant des espèces auxiliaires dans l'arbre d'espèces, chacune d'elles représentant une copie différente du gène. Les deux derniers cas constituent une voie prometteuse pour de futures recherches. Selon les résultats des simulations, la dissimilarité de bipartitions qui prétend comparer plutôt la qualité des bipartitions d'arbres, et non leur quantité comme le fait la distance RF, est beaucoup plus appropriée que RF pour trouver des scénarios optimaux d'opérations SPR (i.e., THG) servant à réconcilier des phylogénies d'espèces et de gènes données.

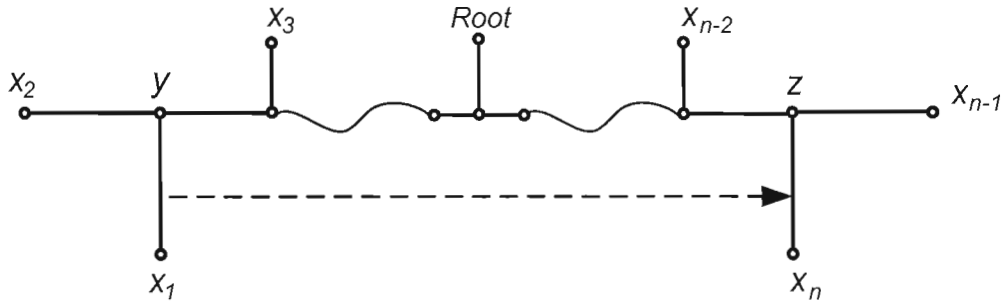


Figure 3.18 Une opération SPR transformant l'arbre d'espèces *T* en l'arbre de gène *T'*. La distance RF entre les arbres *T* et *T'* est égale à sa valeur maximale 2*n*-6 alors que la distance SPR entre *T* et *T'* est égale à 1.

En outre, la contrainte de sous-arbres (figure 3.2) considérée dans *HGT-Detection* offre plusieurs avantages importants (Makarenkov et al., 2006 ; Boc et al., 2010a). Premièrement, l'ordre de THG inférés sous cette contrainte est opposé à leur ordre d'évolution réel. La plupart des programmes de détection (e.g., *LatTrans*) ne produisent pas de THG dans un ordre strict d'évolution. Deuxièmement, la contrainte de sous-arbres prend soin de toutes les contraintes d'évolution nécessaires (figure 3.1 ; voir aussi Maddison, 1997 ou Page et Charleston, 1998) telles que les transferts sur la même lignée ou des transferts croisés. Toutes ces contraintes sont prises en compte automatiquement quand on utilise la contrainte de sous-arbres car les deux sous-arbres impliqués dans un THG doivent être présents dans l'arbre de gène autant que le nouveau sous-arbre qu'ils forment après le transfert (figure 3.2). Troisièmement, l'utilisation de cette contrainte permet de réduire la taille du problème à chaque étape de l'algorithme, en contractant les sous-arbres identiques dans des phylogénies

d'espèces et de gène et en le remplaçant par des arêtes auxiliaires uniques. Quatrièmement, les deux derniers arguments offrent aussi un important gain en temps d'exécution pour ce problème connu comme computationnellement très difficile. L'importance d'un tel gain s'exhibe particulièrement quand il s'agit de calculer le support de bootstrap des transferts horizontaux.

Comme toute méthode d'analyse phylogénétique, l'algorithme de détection des THG décrit dans cette thèse est contraint à un nombre d'artéfacts qui affectent généralement l'inférence phylogénétique. Les principaux d'entre eux sont l'attraction des longues branches, les taux d'évolution inégaux et les situations quand des THG coïncident, ou presque, avec des évènements de spéciation. Dans le futur, il sera important d'investiguer en détail l'impact des ces artéfacts sur les performances des algorithmes de détection des transferts horizontaux. Dans certain cas, notre algorithme peut échouer à obtenir un scénario optimal de THG ou peut inférer des THG allant dans la direction opposée. Ce dernier cas apparait quand une paire de THG, qui diffèrent seulement par leur direction, mène aux mêmes réarrangements topologiques de l'arbre d'espèces (e.g., les THG 2 et 3 et leurs opposés sur la figure 3.13). De tels transferts ont habituellement un faible support de bootstrap. Le problème de la non-inférence d'un scénario de THG optimal est typique pour des petits arbres englobant un grand nombre de transferts. Cependant, l'algorithme de recherche exhaustive *LatTrans* (figure 3.8) et l'heuristique *RIATA-HGT* (figure 3.10) ne se débrouillent pas mieux dans de telles situations (notre algorithme surclasse habituellement les deux derniers dans ces conditions). Des simulations Monté-Carlo exhaustives ont été menées dans le but de comparer les quatre mesures considérées (LS, QD, RF et BD) dans le contexte de l'inférence de transferts horizontaux. Ces simulations démontrent que l'algorithme basé sur BD surclasse les trois autres stratégies dans la plupart des circonstances (voir les figures 3.6 et 3.7). La procédure basée sur RF a été prouvée la moins fiable des quatre stratégies. En plus, la procédure basée sur BD a été comparée à l'algorithme exponentiel *LatTrans* (Hallett and Lagergren, 2001) et à une heuristique rapide *RIATA-HGT* (Nakhleh *et al.*, 2005; Than et Nakhleh, 2008) en termes de taux de détection des THG et de temps d'exécution. Tandis que *HGT-Detection* et *LatTrans* présentent des résultats très similaires en termes de recouvrement des THG (voir les figures 3.6 et 3.8), notre algorithme reste beaucoup plus rapide que *LatTrans* (figure 3.9).

D'autre part, la stratégie basée sur BD surclasse *RIATA-HGT* en termes de détection des transferts horizontaux et de temps d'exécution (figure 3.10). Mentionnons que le nouvel algorithme peut être particulièrement utile pour valider les transferts obtenus par bootstrap. Trois façons d'effectuer la validation des THG par bootstrap ont été suggérées selon des données disponibles. Le calcul du support de bootstrap des THG peut être effectué prenant en compte la robustesse de l'arbre d'espèces, celle de l'arbre de gène, ainsi que le taux de THG obtenus dans tous les scénarios de coût minimal trouvés pour une paire d'arbres d'espèces et de gène (voir les Formules 2 et 3).

La nouvelle version du programme *T-Rex* (Makarenkov, 2001), incluant l'algorithme décrit pour la prédiction et la validation des THG de même que les jeux de données des exemples discutés est disponible à l'adresse suivante : http://www.trex.uqam.ca. Le code source de principales applications incluses dans *T-Rex* est présenté dans l'Annexe C.

[Cette page a été laissée intentionnellement blanche]

CHAPITRE IV

MODÈLE DU TRANSFERT HORIZONTAL PARTIEL

4.1 Introduction

Dans le chapitre précédent, nous avons décrit un nouvel algorithme pour la détection et la validation des transferts horizontaux de gènes. Cet algorithme s'applique au modèle du transfert complet qui assume soit que le gène transféré supplante le gène orthologue entier dans le génome receveur, soit, si le gène transféré est absent du génome receveur, qu'il lui est ajouté (Boc *et al.*, 2010a). Le second modèle, celui du transfert partiel (Makarenkov *et al.*, 2006 et 2008; Boc et Makarenkov 2011) implique la formation des gènes mosaïques. Un gène mosaïque est un allèle (i.e., copie d'un gène) acquis à travers une transformation ou une conjugaison (e.g., à partir de bactéries différentes) et une intégration subséquente, par le biais d'une recombinaison intragénique, dans l'allèle original de l'hôte (Hollingshead *et al.*, 2000, Zhaxybayeva *et al.*, 2004). Ce dernier modèle est une généralisation du premier, car il considère des sous-ensembles de la séquence du gène transféré horizontalement.

Alors que plusieurs méthodes ont été proposées pour l'identification et la validation des THG complets (Hein, 1990; von Haeseler et Churchill, 1993; Page 1994; Mirkin *et al.*, 1995, Maddison, 1997; Lawrence et Ochman, 1997; Charleston, 1998; Hallett et Lagergren, 2001, Boc et Makarenkov, 2003; MacLeod *et al.*, 2005; Nakhleh *et al.*, 2005; Tsirigos et Rigoutsos, 2005; Beiko et Hamilton, 2006; Jin *et al.*, 2006 et 2007; Linz *et al.*, 2007; Than et Nakhleh 2008; Boc *et al.*, 2010), seulement deux méthodes traitent du problème de l'inférence de THG partiels et prédisent les origines des gènes mosaïques (Denamur *et al.*, 2000 et Makarenkov *et al.*, 2006). Cependant, ces deux travaux ne traitent pas la validation des transferts partiels obtenus et n'incluent pas de simulations Monté-Carlo pour tester les

performances des algorithmes dans différentes situations pratiques. Dans les faits, aucune méthode fiable pour l'identification des gènes mosaïques et des transferts horizontaux de gènes (THG) partiels correspondants n'a été proposée jusqu'à ce jour, alors que l'identification et la validation effective des gènes mosaïques est un des défis majeurs en biologie computationnelle.

Nous proposons, dans ce chapitre deux algorithmes pour la prédiction de THG partiels. Le premier se base sur l'optimisation des distances par les moindres carrés pour évaluer les portions de séquences transférées (Makarenkov *et al.*, 2006 et 2008). Un modèle d'optimisation et un algorithme sont alors présentés. Le deuxième modèle se base sur l'algorithme décrit dans (Boc et Makarenkov, 2011) associé au principe de fenêtres coulissantes. Cette approche permet d'identifier des transferts partiels en se déplaçant le long d'un alignement de séquences multiples. Dans ce second algorithme, une procédure de bootstrap est utilisée pour évaluer le support de chaque transfert génétique prédit. Les simulations complètes ont aussi été réalisées pour tester la capacité de cet algorithme d'identifier correctement les transferts partiels en fonction du nombre de transferts et du nombre d'espèces (i.e., la taille de l'arbre). Dans la section résultats, cet algorithme est appliqué pour inférer des THG partiels dans le contexte de l'évolution des gènes *rbcL* (originalement considéré par Delwiche et Palmer, 1996) et *mutU* (originalement considéré par Denamur *et al.*, 2000). La méthode proposée peut aussi être utilisée pour confirmer ou infirmer des transferts de gènes complets inférés pour une paire d'arbres d'espèces et de gène. De plus, le nouvel algorithme peut être utilisé à l'échelle génomique pour évaluer les taux de THG (complets et partiels) entre des génomes alignés. L'algorithme proposé est inclus dans le paquet *T-REX* (Makarenkov, 2001) disponible à l'URL : www.trex.uqam.ca. Mais tout d'abord, il est important de comprendre le rôle et l'importance des gènes mosaïques, ainsi que leur rapport avec le transfert horizontal de gènes.

4.2 Les gènes mosaïques

Les bactéries et les archées s'adaptent à différentes conditions environnementales via l'acquisition de gènes mosaïques (Davison, 1999; Gogarten *et al.*, 2002). Le terme "mosaïque" découle de la configuration des blocs entrecoupés de séquences ayant des histoires d'évolution différentes, mais se trouvant combinés dans l'allèle résultant suite à des

évènements de recombinaison (figure 4.1). Les segments recombinés peuvent être dérivés d'autres souches d'espèces similaires ou d'espèces distantes (Hollingshead *et al.*, 2000, Gogarten *et al.*, 2002). Un gène mosaïque est composé de séquences polymorphes identiques à l'allèle original dans certaines parties du gène, mais contenant, d'autre part, des parties dérivées de l'ADN intégré. Quand l'ADN entrant est très différent de l'ADN hôte, les gènes mosaïques peuvent exprimer des protéines avec de nouveaux phénotypes (e.g., dans le cas où l'ADN du donneur dérive d'une espèce assez distance ou d'un gène différent). Il existe une évidence biologique que les gènes mosaïques soient générés constamment dans les populations d'organismes transformables, et très probablement dans tous leurs gènes (Maiden, 1998). Les gènes mosaïques ont aussi été observés chez des bactéries non-transformables, mais à une fréquence plus faible. Par exemple, chez les espèces de *Neisseria*, les allèles mosaïques ont été reportés pour plusieurs gènes, comprenant ceux encodant des antigènes de surface, la protéase IgA et des cibles antibiotiques (Maiden, 1998; Hollingshead *et al.*, 2000). Un des exemples de gènes mosaïques le plus typique, résultant de transferts horizontaux entre les espèces, est celui du gène qui encode les protéines de liaison résistantes à la pénicilline (PBP) chez *Streptococcus pneunomiae*. Ces protéines sont des cibles létales des B-Lactams de la pénicilline (Maiden, 1998, Claverys *et al.*, 2000). Les pneumocoques, capables de transferts horizontaux entre les espèces, devraient même subir, en toute vraisemblance, des THG fréquents à l'intérieur des espèces (i.e., entre différentes souches) qui contribueraient au développement des allèles mosaïques (Hollingshead *et al.*, 2000).

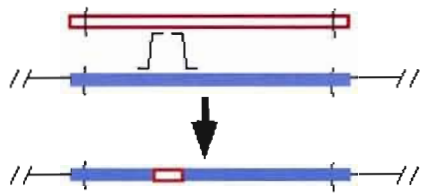


Figure 4.1 Un gène mosaïque incluant une sous-séquence (en blanc) provenant d'une autre espèce.

4.3 Premier modèle d'inférence des transferts partiels

Dans un arbre phylogénétique, il y a toujours un chemin unique connectant une paire de nœuds. L'ajout d'une arête de THG crée un chemin supplémentaire entre certains nœuds. La figure 4.2 illustre le cas où la distance d'évolution entre les taxons i et j peut être affectée par l'ajout du THG (b,a) représentant le transfert partiel du gène donné de b vers a. Il est pertinent d'assumer que le THG entre b et a peut affecter la distance d'évolution entre les taxons i et j, si et seulement si, le point de destination a est situé sur le chemin entre i et la racine de l'arbre ; la position de j est supposée être fixe. Alors, dans la phylogénie réticulée T de la figure 4.2, la distance d'évolution $d_1(i,j)$ entre les taxons i et j peut être calculée comme suit :

$$d_1(i,j) = (1 - \alpha)\, d(i,j) + \alpha\, (d(i,a) + d(j,b)), \qquad (1)$$

où α indique la fraction du gène transféré, inconnue à l'avance et d est la distance entre les nœuds dans l'arbre d'espèces avant l'ajout du transfert (b,a) (voir Makarenkov *et al.*, 2008).

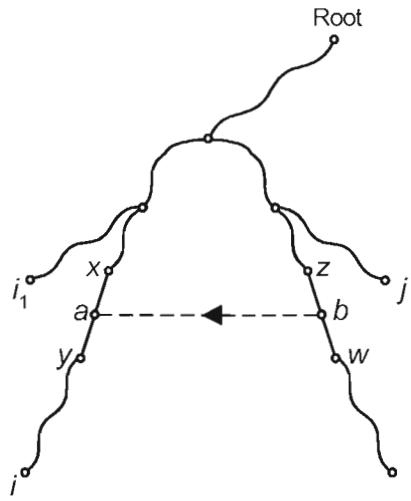


Figure 4.2 La situation où le transfert (b,a) affecte la distance $d(i,j)$, mais pas la distance $d(i_1,j)$.

Par contre, la distance entre les taxons i_1 et j (figure 4.2) ne doit pas être affectée par l'ajout de l'arête (b,a). La figure 4.3 illustre les autres cas où l'ajout d'une arête ne doit pas affecter la longueur du chemin entre i et j.

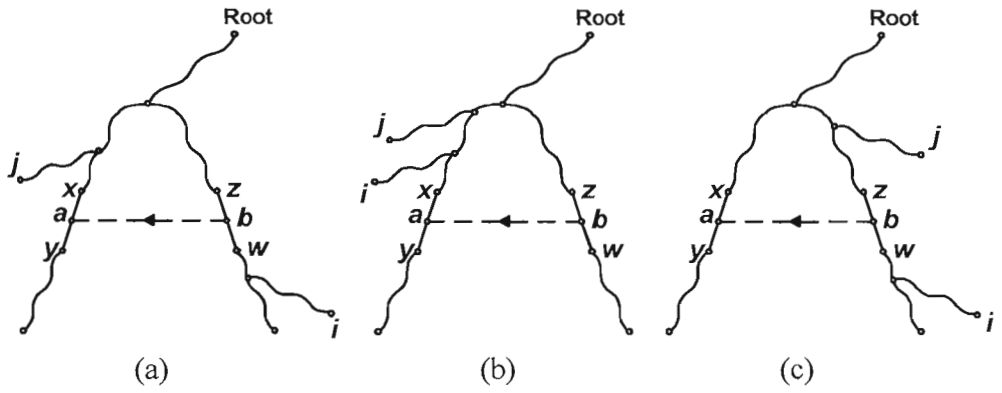


Figure 4.3 Situations où la distance évolutive entre i et j ne change pas après l'ajout de la nouvelle arête (b,a).

La fonction des moindres carrés, à minimiser, Q, avec un vecteur de longueurs d'arêtes $\boldsymbol{l}$ dans T et une fraction inconnue du gène transféré α, est comme suit :

$$Q(L,\alpha) = \sum_{ij \in S} ((1-\alpha) \sum_{k \in path(ij)} l_{ij}^k + \alpha(\sum_{k \in path(ia)} l_{ia}^k + \sum_{k \in path(jb)} l_{jb}^k) - \delta(i,j))^2 + \sum_{ij \notin S} (\sum_{k \in path(ij)} l_{ij}^k - \delta(i,j))^2 \rightarrow min, \quad (2)$$

où $\delta(i,j)$ est la valeur de la dissimilarité initiale entre i et j, l_{ij}^k est la longueur de l'arête k du chemin (ij) dans T, α est la fraction du gène transférée ($0 \leq \alpha \leq 1$) et S est l'ensemble des paires de taxons $\{ij\}$ tels que le transfert (b,a) peut affecter la distance d'évolution entre eux. Pour montrer que le problème d'optimisation par les moindres carrés du transfert partiel de gène, est NP-difficile, le problème suivant peut être énoncé :

Données : *L'arbre phylogénétique d'espèces T (avec la matrice de distances d associée à T sur l'ensemble de taxons X), la dissimilarité de gène* $\boldsymbol{\delta}$ sur *X et une valeur fixe non-négative* ε.

Problème : *Trouver le nombre minimal de transferts partiels k tels que :*

$$Q = \sum_i \sum_j (d_k(i,j) - \delta(i,j))^2 \leq \varepsilon, \tag{3}$$

où $d_k(i,j)$ *est la distance entre les taxons i et j, calculée en utilisant les Formules 1 et 2 dans le réseau phylogénétique* T_k *obtenu à partir de T après l'ajout de k transferts de gènes partiels.*

Théorème 1. *Le problème du nombre minimal de transferts partiels de gène (MNPGT) est NP-difficile.*

La preuve de ce théorème est basée sur une réduction en un temps polynomial du *Problème de Transferts des Sous-arbres* (le problème PTS) qui consiste à trouver le nombre minimal de transferts de gène complets pour transformer un arbre d'espèces T donné en un arbre de gène T' donné. Le problème PTS est identique à celui de l'ajout à T du nombre minimal de transferts de gènes complets tels que $Q = \sum_i \sum_j (d_k(i,j) - \delta(i,j))^2 \leq 0$ (i.e., le cas où $\varepsilon = 0$ est considéré), où $d_k(i,j)$ est la distance entre i et j dans l'arbre phylogénétique (i.e., un cas particulier de réseau phylogénétique). Ici, l'arbre T_k est obtenu à partir de T par l'ajout de k transferts complets (i.e., un cas particulier d'un transfert partiel) et $\delta(i,j)$ est la matrice de distances associée à l'arbre de gène T'. □

Plusieurs contraintes temporelles importantes doivent être incorporées dans ce modèle, en plus de celles qui sont déjà prises en compte dans le modèle du transfert complet, pour identifier les interactions entre les THG partiels qui ne sont pas éligibles d'un point de vue de l'évolution. Certaines de ces contraintes, ont été initialement indiquées par Page et Charleston (1998a et b). Par exemple, les transferts croisés entre deux lignées (figure 4.4) doivent être interdits.

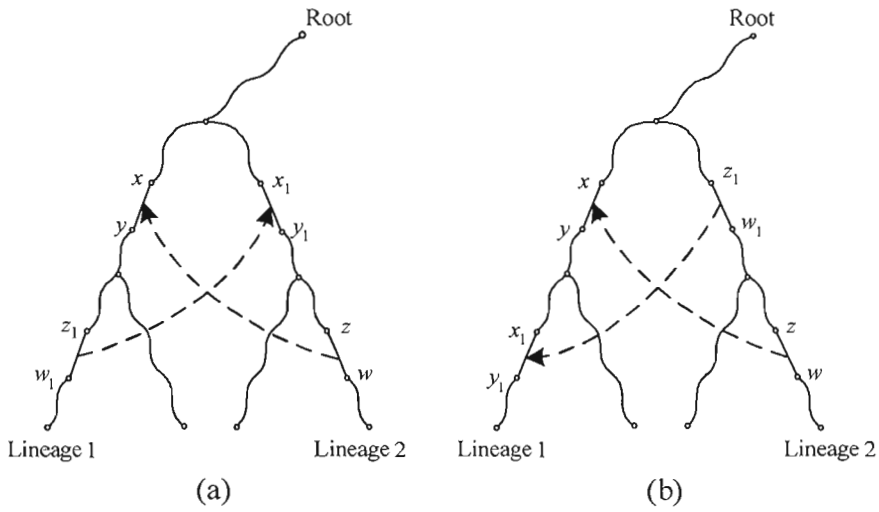


Figure 4.4 Les transferts croisés doivent être interdits.

Notons que la règle illustrée dans la figure 4.4a est automatiquement prise en compte dans le modèle de transferts de gènes complets, où ces violations seraient équivalentes à la violation de la contrainte de la même lignée (voir Page et Charleston 1998). Par exemple, (figure 4.4a), le THG de (z,w) vers (x,y) ne peut pas être suivis par le transfert allant de (z_1,w_1) à (x_1,y_1) qui sera localisé sur la même lignée (lignée 2). Nous avons aussi identifié deux cas, où la distance d'évolution entre les taxons i et j peut être affectée par des transferts multiples (figure 4.5a et b) ; et, deux cas, où la distance ne doit pas être affectée par ces transferts (figure 4.5c et d). Ne pas prendre en compte ces contraintes peut résulter en des transferts mutuellement incompatibles.

Supposons qu'un transfert partiel entre les arêtes (z,w) et (x,y) (i.e., de b à a dans la figure 4.2) de l'arbre d'espèces T ait lieu. Les longueurs de toutes les arêtes dans T sont réévaluées selon les moindres carrés après l'ajout de l'arête (b,a), alors que la longueur de (b,a) est supposée être 0.

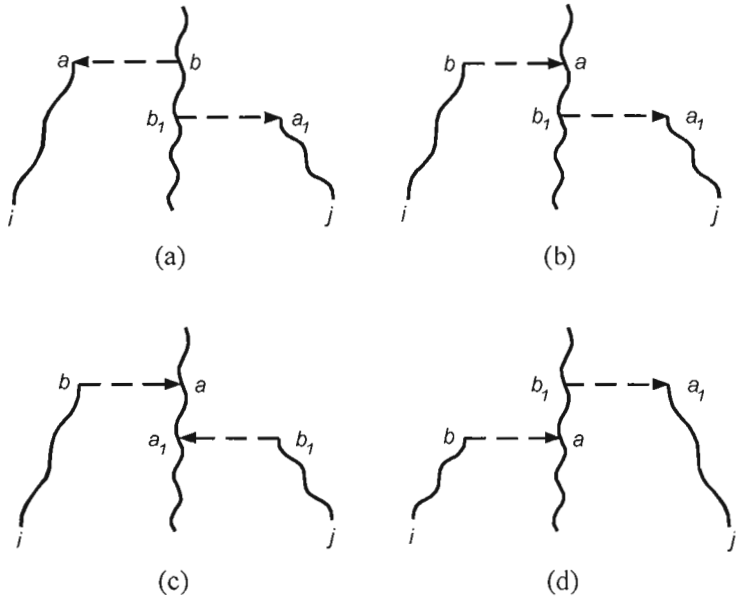


Figure 4.5 La distance entre les feuilles *i* et *j* peut être affectée par les deux transferts présentés dans les portions (a) et (b) de la figure. Cette distance ne peut pas être affectée dans les cas présentés dans les portions (c) et (d) de la figure.

Pour réévaluer les longueurs des arêtes de T, nous avons tout d'abord fait une supposition concernant la valeur du paramètre α (Equation 1), indiquant la fraction du gène qui a été transférée. Ce paramètre peut être estimé soit en comparant des séquences correspondant aux sous-arbres enracinés par les nœuds y et w, soit par le test de différentes valeurs α dans le problème d'optimisation.

En fixant le paramètre α, nous réduisons à un système linéaire le système d'équations établissant la correspondance entre les distances génétiques expérimentales et les distances dans le réseau de transferts. Ce système, qui a généralement plus de variables (i.e., longueurs des arêtes de T) que d'équations (i.e., paires de distances dans T ; le nombre d'équations est toujours $n(n\text{-}1)/2$ pour n taxons), peut être résolu en utilisant l'approximation par les moindres carrés. Voyons maintenant comment le problème d'approximation peut être formulé et efficacement résolu.

Soit $\mathbf{A}_\alpha$ la matrice de dimension $n(n\text{-}1)/2$ x m, chaque ligne étant une paire de taxons de X, où n est le nombre de taxons et m est le nombre d'arêtes dans T. La valeur $a_{ij,e}$ de cette matrice, correspondant à la paire de taxons ij et à l'arête e, est égale soit à 1, soit à α, soit à 1-α (si l'arête e se trouve sur le chemin (ij) dans T), ou est égale à 0 sinon. Soit $\boldsymbol{\ell}$ le vecteur de longueurs d'arêtes à m éléments et $\mathbf{d}$ le vecteur de distances de gènes à $n(n\text{-}1)/2$ éléments.

En fixant la valeur de α (e.g., les valeurs 0, 0.1, 0.2, ..., et 1.0 peuvent être testées), nous obtenons un système d'équations linéaires avec $n(n\text{-}1)/2$ équations et m inconnues : $\mathbf{A}_\alpha \times \boldsymbol{\ell} = \mathbf{d}$. Quand $n \geq 4$, ce système a plus d'équations que d'inconnues. Il peut donc être résolu par approximation de la façon suivante :

$$(\mathbf{A}_\alpha \times \boldsymbol{\ell} - \mathbf{d})^2 \rightarrow \min. \tag{4}$$

Après avoir pris le gradient, nous avons:

$$\mathbf{A}_\alpha^t \times (\mathbf{A}_\alpha \times \boldsymbol{\ell} - \mathbf{d}) = 0 \tag{5}$$

À la suite de manipulations algébriques, nous obtenons :

$$\mathbf{A}_\alpha^t \times \mathbf{A}_\alpha \times \boldsymbol{\ell} = \mathbf{A}_\alpha^t \times \mathbf{d} \tag{6}$$

Alors, nous avons : $\mathbf{B} \times \boldsymbol{\ell} = \mathbf{c}$, où $\mathbf{B}$ est une matrice ($m \times m$) et $\mathbf{c}$ est un vecteur à m composantes.

Suivant Barthélemy et Guénoche (1988) et Makarenkov et Leclerc (1999), nous appliquons une méthode de Gauss-Seidel légèrement modifiée pour résoudre le système ci-dessus. La méthode consiste à décomposer $\mathbf{B}$ dans sa diagonale (Δ), sa composante triangulaire supérieure stricte ($-\mathbf{F}$) et sa composante triangulaire inférieure stricte ($-\mathbf{E}$):

$$\mathbf{B} = \begin{pmatrix} b_{11} & b_{12} & \cdots & b_{1m} \\ b_{21} & b_{22} & \cdots & b_{2m} \\ \cdots & \cdots & \cdots & \cdots \\ b_{m1} & b_{m2} & \cdots & b_{mm} \end{pmatrix} = \begin{pmatrix} & & -\mathbf{F} \\ & \Delta & \\ -\mathbf{E} & & \end{pmatrix} = \Delta - \mathbf{E} - \mathbf{F}. \tag{7}$$

Alors, nous appliquons la procédure itérative suivante :

$$\Delta \times \ell^{(k+1)} = \mathbf{E} \times \ell^{(k+1)} + \mathbf{F} \times \ell^{(k)} + \mathbf{c}, \qquad (8)$$

qui nous permet de calculer graduellement les composantes du vecteur $\ell(j)^{(k+1)}$, correspondant aux longueurs des arêtes de T à la k+1-ème itération, à partir de $\ell(j)^{k}$. Si la valeur calculée de $\ell(j)^{(k+1)}$ est négative, elle est remplacée par la valeur de 0. Cette opération est équivalente à la projection sur le cône $\ell \geq 0$, qui assure une solution appropriée (i.e., non-négative).

L'équation exacte utilisée dans cette méthode est la suivante, pour tous $j = 1, 2, \ldots, m$:

$$\ell(j)^{(k+1)} = (- (\sum_{j+1 \leq i \leq m} b_{ij} \ell(j)^{(k)}) - (\sum_{1 \leq i \leq j-1} b_{ij} \ell(j)^{(k+1)}) + c_j) / b_{jj}. \qquad (9)$$

Les principales étapes de l'algorithme pour la détection des transferts de gène partiels peuvent être énoncées comme suit :

Étape préliminaire. Cette étape correspond à l'étape préliminaire discutée dans le contexte du modèle du transfert complet. Elle consiste à inférer les phylogénies d'espèces et de gène, notées respectivement T et T', dont les feuilles sont étiquetées par le même ensemble X de n taxons. Nous utilisons le critère des moindres carrés comme unique critère d'optimisation quand nous modélisons les transferts horizontaux partiels car les trois autres critères considérés (i.e., RF, QD et BD) s'appliquent uniquement à des topologies d'arbres.

Étape 1. Tester toutes les connexions entre les paires d'arêtes dans l'arbre T. Pour chaque THG satisfaisant les contraintes d'évolution, effectuer les opérations suivantes :

a) Fixer la valeur de la fraction transférée du gène α (e.g., on peut essayer à chaque tour les valeurs de 0, 0.1, 0.2, ..., et 1.0). Calculer les longueurs optimales ℓ des arêtes dans l'arbre d'espèces (ou dans le réseau, commençant par l'étape 2) T en utilisant la méthode itérative de Gauss-Seidel (Formules 8 et 9).

b) Retourner au système d'équations original : $\mathbf{A}_a \times \ell = \mathbf{d}$. Fixer les valeurs du vecteur ℓ trouvées en utilisant la méthode de Gauss-Siedel et résoudre le problème par les moindres carrés en considérant comme inconnu le paramètre α.

c) Fixer la valeur optimale de α trouvée et répéter le calcul jusqu'à ce que les deux paramètres inconnus convergent vers une solution.

Toutes les paires d'arêtes éligibles (i.e., satisfaisant les contraintes d'évolution) dans T peuvent être traitées de cette façon. Le THG produisant la plus petite valeur du coefficient des moindres carrés, Q, et satisfaisant les contraintes d'évolution définies, est sélectionné pour l'ajout dans l'arbre d'espèces T, le transformant ainsi en un réseau phylogénétique.

Étape 2..*k*. Exécuter l'algorithme jusqu'à ce qu'un nombre fixe de transferts partiels, k, soit trouvé et ajouté à T ou que la valeur de Q soit plus petite qu'un seuil préétabli ε.

La complexité algorithmique de cette méthode est $O(kn^5)$ pour ajouter k transferts horizontaux de gène partiels dans un arbre phylogénétique d'espèces à n feuilles.

4.4 Deuxième modèle d'inférence des transferts partiels

Dans cette section nous décrivons les principales propriétés de notre deuxième approche algorithmique servant à inférer des THG partiels. Les principales étapes de l'algorithme, qui cherche à produire un scénario optimal de transferts partiels d'un gène donné pour un groupe d'espèces considéré, sont résumées ci-dessous. Une validation par bootstrap est effectuée pour chaque transfert partiel hypothétique et seuls les transferts avec les valeurs de bootstrap les plus élevées sont inclus dans la solution finale. Une procédure de fenêtre coulissante est utilisée pour tester différents fragments de l'alignement de séquences multiples (ASM). Notons qu'une approche utilisant une fenêtre coulissante a été précédemment utilisée pour détecter les recombinaisons (Ray 1998; Paraskevis *et al.*, 2005; Lee et Sung, 2008), mais aucun de ces travaux ne traite le problème du transfert de gènes partiels. Un algorithme pour l'identification des transferts complets est utilisé à chaque étape pour réconcilier l'arbre d'espèces donné et l'arbre de gène partiel inféré à partir des séquences situées à l'intérieur de la fenêtre coulissante (Boc et Makarenkov, 2011).

4.4.1 Algorithme pour détecter des transferts partiels

Étape préliminaire. Soit X un ensemble d'espèces, ASM un alignement de séquences multiples de longueur l, et $S_{i,j}$ le fragment de l'ASM, étant analysé, et situé entre les sites i et j, où $1 <= i < j <= l$. Nous définissons aussi la taille de la fenêtre coulissante w ($w = j - i$) et la taille du pas de progression s. Inférons l'arbre phylogénétique d'espèces T. Usuellement, un arbre basé sur les caractères morphologiques ou sur une molécule qui est supposée être réfractaire aux transferts horizontaux de gènes (e.g., 16S sRNA ou 23S sRNA) joue le rôle de l'arbre d'espèces. L'arbre T doit être enraciné en respectant les hypothèses d'évolution connues. Si aucune hypothèse biologique plausible pour enraciner T n'est disponible, les stratégies d'outgroup ou de point médian peuvent être utilisées. L'enracinement de l'arbre est important car la racine permet de prendre en compte les règles d'évolution (Maddison 1997, Hallett et Lagergren 2001) qui doivent être respectées lorsqu'on infère des transferts horizontaux. Fixons la taille de la fenêtre coulissante w et la taille du pas s (dans nos expériences, les tailles de fenêtres égales à $l/5$, $l/4$, $l/3$, $l/2$ et le pas de progression de 10 sites ont été utilisés).

Étape k. Fixons la position de la fenêtre coulissante dans l'intervalle $[i,j]$, où $i = 1 + s(k - 1)$ et $j = i + w$ (voir la figure 4.6). Si $i + w > l$, alors $j = l$. Inférons un arbre de gène partiel T' caractérisant l'évolution du fragment de l'ASM localisé dans l'intervalle $[i,j]$. Dans cette étude, la méthode *PHYML* (Guindon et Gascuel, 2003) a été utilisée pour inférer les arbres de gène partiels. Appliquons un algorithme de détection pour inférer un scénario de THG partiels associé à l'intervalle $[i, j]$. Ici, nous avons utilisé l'algorithme *HGT-Detection* décrit dans le chapitre III pour inférer des transferts complets, mais n'importe quel autre algorithme pourrait être utilisé à sa place. Cet algorithme de détection des THG est plus rapide et dans la plupart des cas aussi précis que les populaires algorithmes *LatTrans* (Hallett and Lagergren, 2001) et *RIATA-HGT* (Nakhleh *et al.*, 2005). La dissimilarité de bipartitions (Makarenkov et al., 2007; Boc et al., 2010a) a été utilisée comme critère d'optimisation. Une procédure pour évaluer la fiabilité des transferts partiels obtenus (i.e., support de bootstrap) a aussi été développée. Une telle procédure de bootstrap prend en compte l'incertitude des arbres de gène partiels ainsi que le nombre de fois qu'un transfert donné apparait dans tous les scénarios de coût minimal obtenus pour les arbres d'espèces et de gène donnés (voir les

Formules 2 et 3 dans le chapitre III). Parmi les THG partiels obtenus, nous retenons seulement ceux ayant un score de bootstrap significatif.

Étape finale. Établissons une liste des THG partiels prédits. Identifions les intervalles entrelacés donnant lieu à des transferts partiels identiques (i.e., les mêmes donneurs et receveurs et la même direction). Ré-exécutons l'algorithme de détection des THG pour tous les intervalles entrelacés (considérant leur longueur totale dans chaque cas) produisant les THG partiels identiques. Si ces THG partiels sont trouvés à nouveau (c'était habituellement le cas dans nos expériences) pour le fragment de séquences situé dans les intervalles entrelacés, évaluons leur support de bootstrap et, selon le support obtenu, incluons-les dans la solution finale ou supprimons-les.

La complexité de cet algorithme est comme suit :

$$O(r \times (\frac{(l-w)}{s} \times (C(PhIn) + \tau \times n^4))), \tag{10}$$

où w est la taille de la fenêtre coulissante, s est le pas de progression, $C(PhIn)$ est la complexité de la méthode d'inférence d'arbres phylogénétiques (e.g., *PHYML*) utilisée pour inférer les phylogénies à partir des fragments de séquences dans la fenêtre coulissante, r est le nombre de réplicats dans le bootstrap, n est le nombre d'espèces et τ est le nombre moyen de transferts inférés pour un fragment de séquences de taille w.

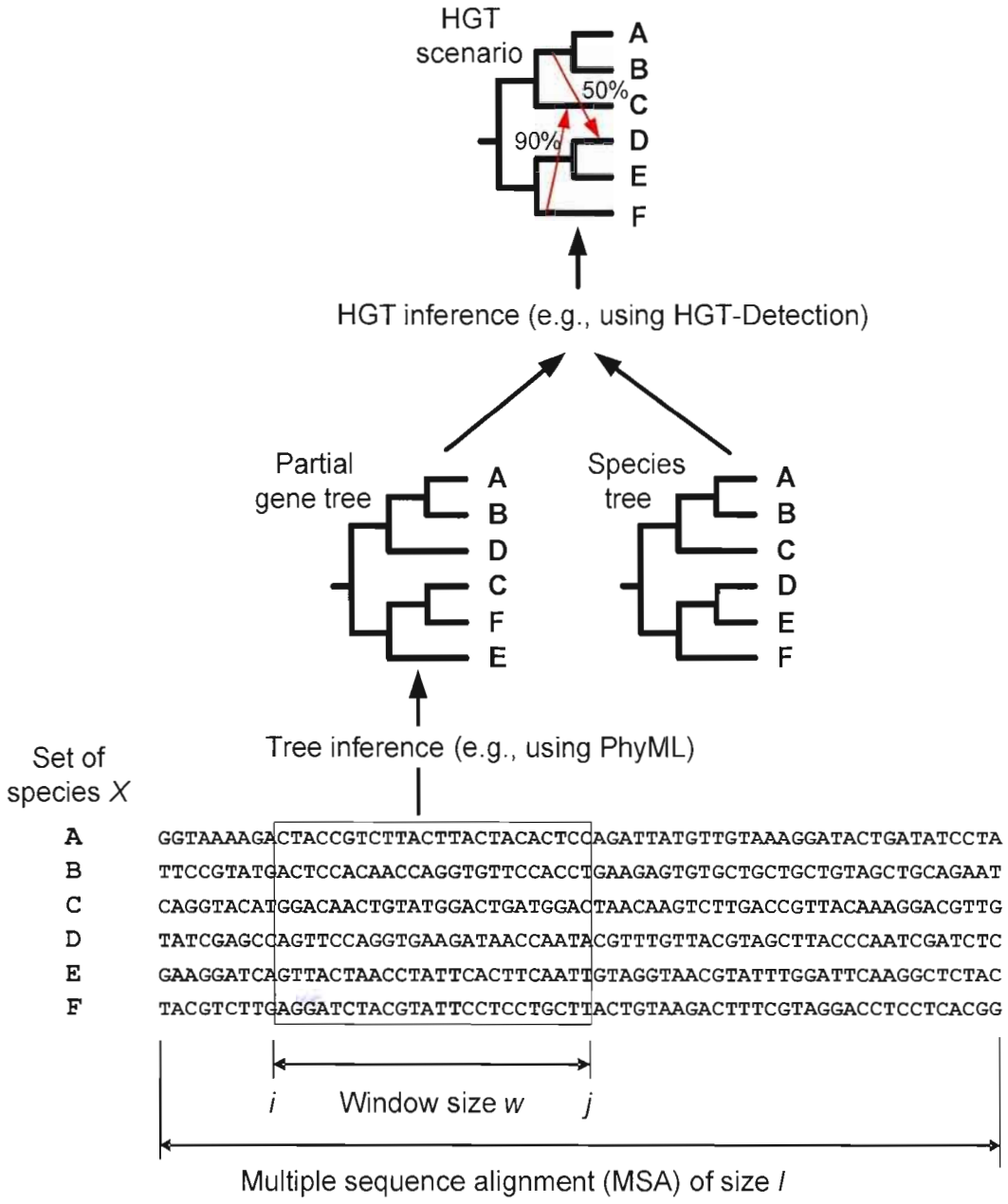


Figure 4.6 L'arbre de gène partiel est inféré en utilisant les séquences situées à l'intérieur de la fenêtre coulissante et l'algorithme de détection des transferts complets (Boc *et al.*, 2010a), incluant l'étape de validation par bootstrap, est alors exécuté.

4.4.2 Simulations Monté-Carlo

Des simulations Monté-Carlo ont été menées pour tester l'efficacité du nouvel algorithme dans le contexte des THG partiels. Nous avons examiné comment l'algorithme proposé se comporte en fonction du nombre d'espèces observées (i.e., la taille de l'arbre) et du nombre de transferts partiels générés. La procédure de simulations inclut les quatre étapes de base suivantes :

Première étape. Des arbres binaires d'espèces avec 8, 16, 32 et 64 feuilles ont été créés en utilisant la procédure de génération d'arbres aléatoires de Kuhner et Felsenstein (1994). Les longueurs des arêtes des arbres ont été générées en utilisant une distribution exponentielle. En suivant l'approche de Guindon et Gascuel (2002), nous avons ajouté un peu de bruit aux arêtes pour modéliser un écart par rapport à l'hypothèse de l'horloge moléculaire. Les arbres aléatoires produits par cette procédure avaient une profondeur de $O(\log(n))$, où n est le nombre d'espèces (i.e., le nombre de feuilles dans un arbre phylogénétique binaire).

Deuxième étape. Nous avons exécuté le programme *SeqGen* (Rambaut et Grassly, 1996) pour générer des alignements de séquences multiples de protéines le long des arêtes des arbres d'espèces construits à la première étape. Le programme *SeqGen* a été utilisé avec le modèle de substitution de protéines JTT (Jones *et al.*, 1992), une distribution Gamma estimée, un nombre de catégories de taux de substitution égale à 4 et une proportion des sites invariables égale à 0. Des séquences de protéines avec 500 acides aminés ont été générées. Ces paramètres ont été sélectionnés dans le but de rendre les paramètres des simulations similaires à ceux utilisés dans la section "Exemples" (voir l'exemple de l'évolution du gène *rbcL*).

Troisième étape. Pour chaque arbre d'espèces T, nous avons généré des arbres de gènes avec le même nombre de feuilles en effectuant des déplacements SPR aléatoires de ses sous-arbres. Un modèle satisfaisant toutes les contraintes d'évolution plausibles a été implémenté pour générer les THG partiels aléatoires. Pour chaque arbre d'espèces, 1 à 5 déplacements SPR aléatoires ont été effectués et différents arbres de gènes T', englobant entre 1 et 5 THG partiels, ont été générés. Pour chaque arbre de gène, les fragments de séquences dans les sous-arbres affectés par les THG ont été régénérés avec *SeqGen*. Nous avons fixé la taille de

chaque séquence transférée à 200 acides aminés. Les alignements de séquences multiples obtenus contenaient donc des blocs de séquences affectées par des THG.

Quatrième étape. Nous avons exécuté l'algorithme pour chaque arbre d'espèces généré et l'alignement de séquences multiples associé qui était affecté par les THG partiels. La taille de la fenêtre coulissante a été fixée à 100, 200, 300, 400, puis 500 acides aminés ; 100 réplicats de chaque arbre de gène partiel T' ont été générés pour évaluer le support de bootstrap des arêtes de T' dans un premier temps, puis le support des THG partiels obtenus dans un deuxième temps. Les arbres qui avaient le support de bootstrap inférieur à 60% ont été retirés de l'analyse. Parmi les THG obtenus, seuls les transferts avec un bootstrap supérieur à 90% ont été retenus dans la solution finale. Finalement, nous avons estimé le taux de détection (les vrais positifs seulement) et le taux de faux positifs en fonction du nombre d'espèces et de transferts générés.

Les performances moyennes obtenues par cet algorithme sont illustrées sur les figures 4.7 et 4.8. Pour chaque ensemble de paramètres (taille de l'arbre, nombre de THG générés), 100 jeux de données répliqués ont été testés. La figure 4.7 (a) montre que les meilleurs taux de détection ont été obtenus pour les arbres de 16 espèces. Les résultats variaient de 100%, pour un transfert, à 88%, pour cinq transferts. En outre, on peut noter à la figure 4.7b que le taux de faux positifs pour les arbres de 16 espèces était habituellement plus petit que 40%. Les mêmes tendances peuvent être observées pour les autres tailles d'arbres. La figure 4.8 met en lumière les différences entre le taux de détection moyen et le taux de faux positifs moyen en fonction du nombre d'espèces. La moyenne ici était calculée à partir des résultats obtenus pour 1 à 5 THG générés. Alors que le taux de détection moyen était toujours supérieur à 70% (79,6% en moyenne), le taux moyen de faux positifs était toujours inférieur à 40% (30,8% en moyenne). Selon des tests additionnels (non décrits ici), ces résultats peuvent être améliorés en ajustant les paramètres des simulations dépendamment de la nature des séquences étudiées. Les résultats des simulations suggèrent que cet algorithme peut être utile pour détecter des THG partiels (i.e., identification des gènes mosaïques). Les meilleurs taux de détection ont été obtenus pour les arbres avec 16 et 32 feuilles. Les plus faibles taux de faux positifs ont été obtenus pour les arbres de 32 et 64 feuilles. Alors qu'en moyenne les taux de détection des THG partiels étaient légèrement plus faibles que ceux

obtenus par les algorithmes *LatTrans* (Hallett et Lagergren, 2001) et *HGT-Detection* pour les transferts complets (voir la figure 3.8), il est important de noter que le problème de détection des THG partiels est beaucoup plus complexe en raison de la forte similarité des fragments de séquences situés dans les blocs d'alignements multiples affectés par des THG et à cause des situations où ces blocs s'entrelacent et cachent de réels transferts de gènes.

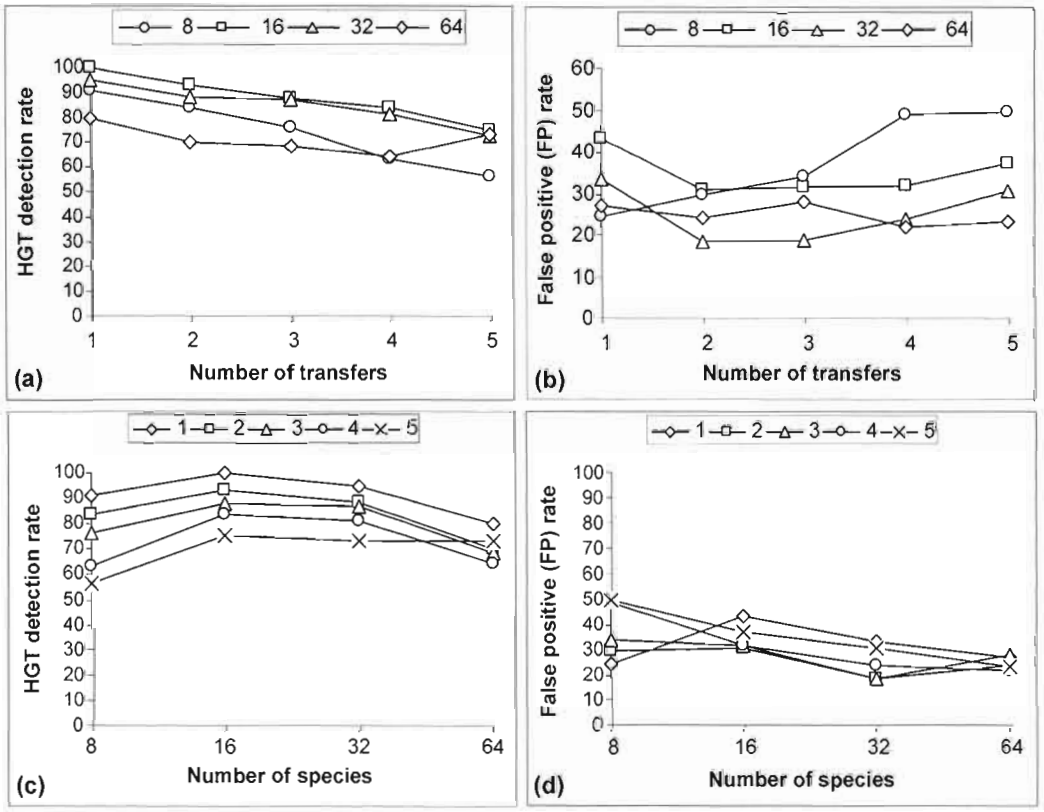


Figure 4.7 Taux de détection et taux de faux positifs en fonction du nombre de feuilles et du nombre de transferts horizontaux partiels. Le taux de détection présenté en fonction du nombre de transferts (a) et du nombre de feuilles (c). Le taux de faux positifs présenté en fonction du nombre de transferts (b) et du nombre de feuilles (d).

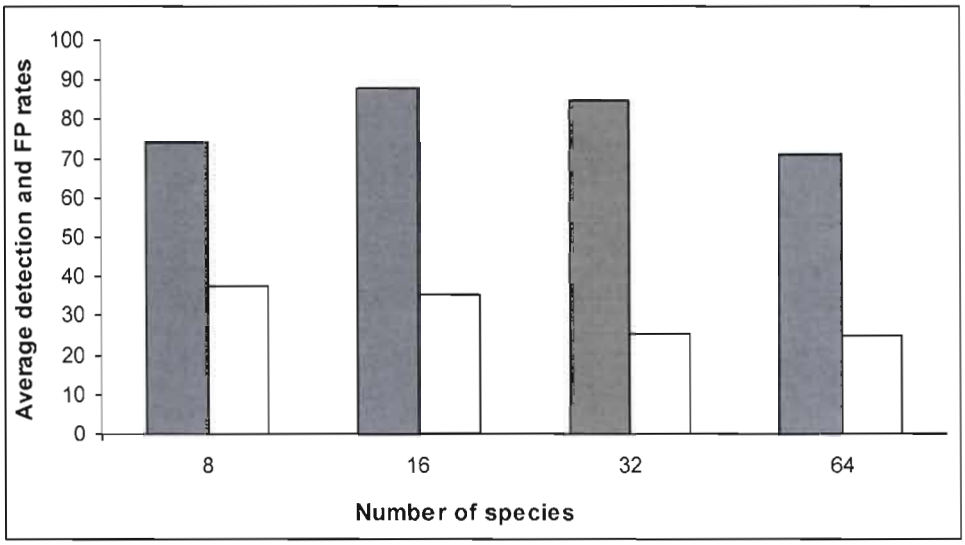


Figure 4.8 Taux moyens de détection (vrais et faux positifs) pour les cas de 1 à 5 transferts partiels générés.

4.4.3 Exemples

4.4.3.1 Détection des transferts partiels du gène *rbcL*

Premièrement, nous avons appliqué le deuxième algorithme de détection de THG partiels à l'analyse des données de protéobactéries, cyanobactéries et plastides originalement examinées par Delwiche et Palmer (1996). Ces auteurs ont discuté l'hypothèse de THG du gène rubisco (*rbcL*) contre l'hypothèse d'une ancienne duplication suivie par des pertes partielles du gène. En utilisant une méthode de maximum de parcimonie, Delwiche et Palmer (1996) ont inféré une phylogénie du gène *rbcL* pour 48 espèces, incluant 42 entrées pour la forme I et 6 entrées pour la forme II du gène rubisco. Ils ont souligné que la classification basée sur le gène *rbcL* contenait plusieurs conflits par rapport à celle basée sur l'ARN ribosomale 16S et d'autres évidences biologiques. Les séquences alignées de la protéine *rbcL* considérées par Delwiche et Palmer et réanalysées dans cette thèse sont disponibles à l'adresse URL suivante : http://www.life.umd.edu/labs/delwiche.

Pour effectuer notre analyse, nous avons considéré 42 des 48 organismes de l'étude originale: toutes les entrées de la forme I de *rbcL* ont été sélectionnées, et les 6 entrées de la

forme II, utilisées par Delwiche et Palmer (1996) pour enraciner l'arbre de gène, ont été écartées. Les espèces *Chromatium* et *Hydrogenovibrio*, ayant deux différentes copies du gène rubisco, notées respectivement, *Chromatium A* et *L*, et *Hydrogenovibrio L1* et *L2*, ont été considérées dans l'étude originale. Donc, dans cet exemple, la phylogénie du gène *rbcL* inclut 42 organismes, alors que la phylogénie d'espèces n'en contient que 40. Il est important de noter que notre algorithme a été adapté pour prendre en compte les arbres d'espèces et de gène ayant un nombre différent de feuilles. L'arbre de maximum de vraisemblance pour le gène *rbcL* inféré en utilisant la méthode *PHYML* (Guindon et Gascuel, 2003) est montré à la figure 4.9. Cet arbre est très similaire à celui obtenu par Delwiche et Palmer (voir la figure 2, 1996).

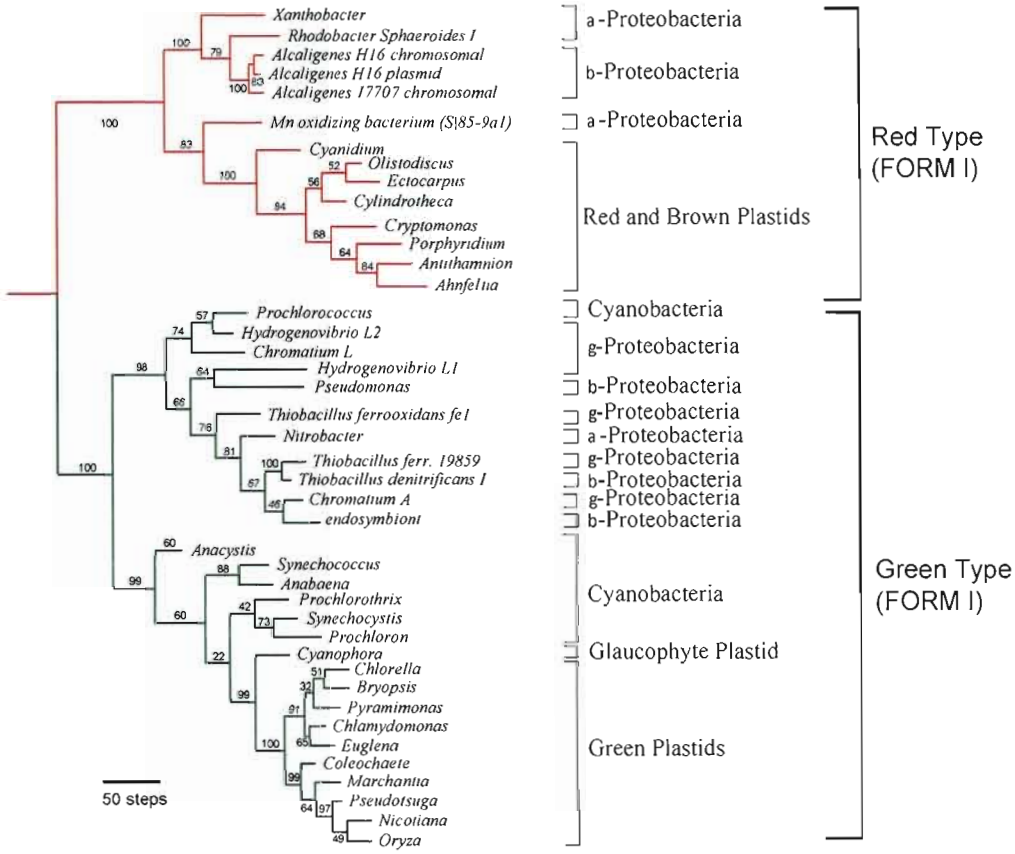


Figure 4.9 La phylogénie du gène *rbcL* pour 42 bactéries et plastides obtenue à l'aide de *PHYML*. La classification basée sur l'ARN ribosomale 16S est indiquée dans la partie droite de la figure. Les scores de bootstrap ont été calculés pour 100 réplicats.

Les organismes *Pseudomonas* et *endosymbiont of Alvinoconcha*, notés comme incertains dans Delwiche et Palmer (1996), ont été récemment classifiés comme des β-protéobactéries. L'arbre d'espèces correspondant (voir la figure 4.10) a été reconstruit et enraciné en utilisant les données appropriées du NCBI (Benson *et al.*, 2009). Comme nous étions plutôt intéressés à identifier les transferts entre les groupes d'organismes, nous avons gardé intacte, en respectant la topologie de l'arbre de gène, la position des organismes appartenant aux mêmes groupes. Par exemple, les topologies des clades des chloroplastes, des cyanobactéries et des algues rouges et brunes sont identiques dans les phylogénies de gène et d'espèces illustrées dans les figures 4.9 et 4.10, respectivement. Un nombre important de conflits topologiques entre les phylogénies d'espèces et de gène peuvent être observés. Par exemple, il existe un grand clade dans l'arbre de gène avec un support de bootstrap de 98% (figure 4.9), incluant une α-protéobactérie, trois β-protéobactéries, six γ-protéobactéries et une cyanobactérie. De tels conflits topologiques peuvent être expliqués soit par un grand nombre de transferts horizontaux (complets ou partiels), soit par une ancienne duplication suivie par des pertes de gènes (ces deux hypothèses ne sont pas mutuellement exclusives ; pour plus de détails, voir Delwiche et Palmer, 1996). Ci-dessous, nous considérons seulement l'hypothèse de THG pour expliquer l'incongruence topologique entre les arbres d'espèces (figure 4.10) et de gène (figure 4.9).

Tout d'abord, nous avons exécuté notre algorithme *HGT-Detection* (voir chapitre III et Boc *et al.*, 2010) pour prédire les THG complets ; la dissimilarité de bipartitions a été utilisée comme critère d'optimisation. Le scénario de coût minimal de 9 THG nécessaires pour réconcilier les phylogénies d'espèces et de gène est présenté à la figure 4.10 (les THG sont représentés par les flèches numérotées). L'optimalité de cette solution a été confirmée par le programme *LatTrans* (Hallett et Lagergren, 2001) qui se base sur une recherche exhaustive. Le support de bootstrap pour les transferts complets a aussi été calculé.

Par la suite, nous avons exécuté le deuxième algorithme de prédiction des THG partiels. Nous avons utilisé des fenêtres coulissantes de tailles 200, 300 et 400 sites avec un pas de progression de 10 sites. Les arbres partiels correspondant aux sous-séquences localisées dans la fenêtre coulissante ont été inférés avec *PHYML* (avec les paramètres indiqués dans la section Simulation Monté-Carlo). Avec la fenêtre de taille 200, le score de

bootstrap moyen des arbres partiels était inférieur à 50% en raison de la forte similarité entre les séquences d'acides aminés considérées. De nouveau, l'algorithme *HGT-Detection* (Boc *et al.*, 2010), avec la dissimilarité de bipartitions comme critère d'optimisation, a été exécuté pour inférer les transferts partiels pour chaque position fixe de la fenêtre coulissante.

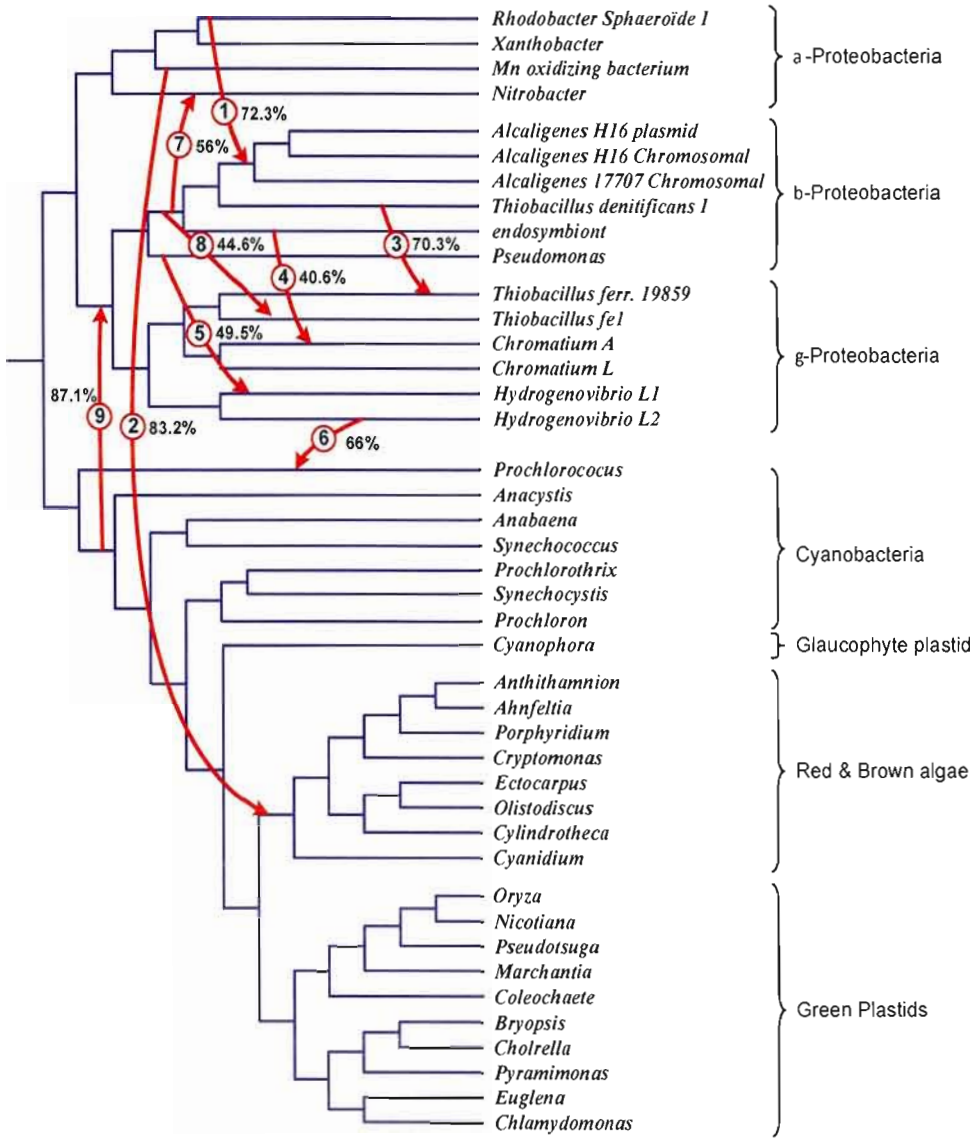


Figure 4.10 Les transferts complets obtenus en appliquant l'algorithme *HGT-Detection*. Un scénario unique de 9 THG (indiqué par les flèches) a été trouvé. C'est le scénario de coût minimal de transferts complets réconciliant les deux arbres.

Comme résultat final, nous avons retenu les 10 transferts partiels illustrés à la figure 4.11 (tous les transferts partiels avec un score de bootstrap inférieur à 60% ont été écartés). Certains de ces transferts correspondent à des transferts complets (i.e., les THG 2, 6 et 9). Le nouvel algorithme pour l'inférence des transferts partiels permet de raffiner les résultats trouvés par un algorithme de détection de transferts complets. Certains transferts complets ont été confirmés comme étant de réels transferts complets (i.e., les THG 2, 6 et 9), d'autres ont été écartés (i.e., les THG 5 et 8 avec un faible score de bootstrap) et d'autres ont été transformés en transferts partiels (i.e., les THG 1, 3, 4 et 7). De plus, trois nouveaux transferts (partiels) ont été trouvés (i.e., les THG 10, 11 et 12). Par exemple, le gène *rbcL* de *Chromatium L*, composé des séquences polymorphes issues d'*Hydrogenovibrio L1* (dans l'intervalle 130:230), d'*Hydrogenovibrio L2* (dans l'intervalle 361:531) et de la séquence originale (dans les intervalles 1:129 et 231:360), est un gène fortement mosaïque. Évidemment, les scores de bootstrap des transferts partiels (trouvés pour une partie de l'ASM) sont supérieurs à ceux obtenus pour les transferts complets équivalents (trouvés pour tout l'ASM).

Les transferts illustrés sur les figures 4.10 et 4.11 incluent un des principaux THG prédits par Delwiche et Palmer (1996, voir la figure 4 et la discussion qui suit) : le transfert entre α-protéobactéries et les algues rouges et brunes (le THG complet 2 avec un score de bootstrap de 83.2%). Le transfert exact entre la cyanobactérie et l'ancêtre des α- et γ-protéobactéries (le THG complet 9 avec un score de bootstrap de 87.1%) n'a pas été prédit par Delwiche et Palmer (1996), mais ces auteurs ont discuté la possibilité d'un transfert proche entre les cyanobactéries et l'ancêtre des γ-protéobactéries. Le scénario de THG partiels n'inclut cependant aucun transfert des γ-protéobactéries vers les α- et β-protéobactéries (HGT prédits par Delwiche et Palmer, 1996). Pour résoudre les conflits topologiques entre les topologies d'espèces et de gène, le scénario obtenu repose sur les THG des β-protéobactéries vers les α- et γ-protéobactéries, et des α-protéobactéries vers les β-protéobactéries.

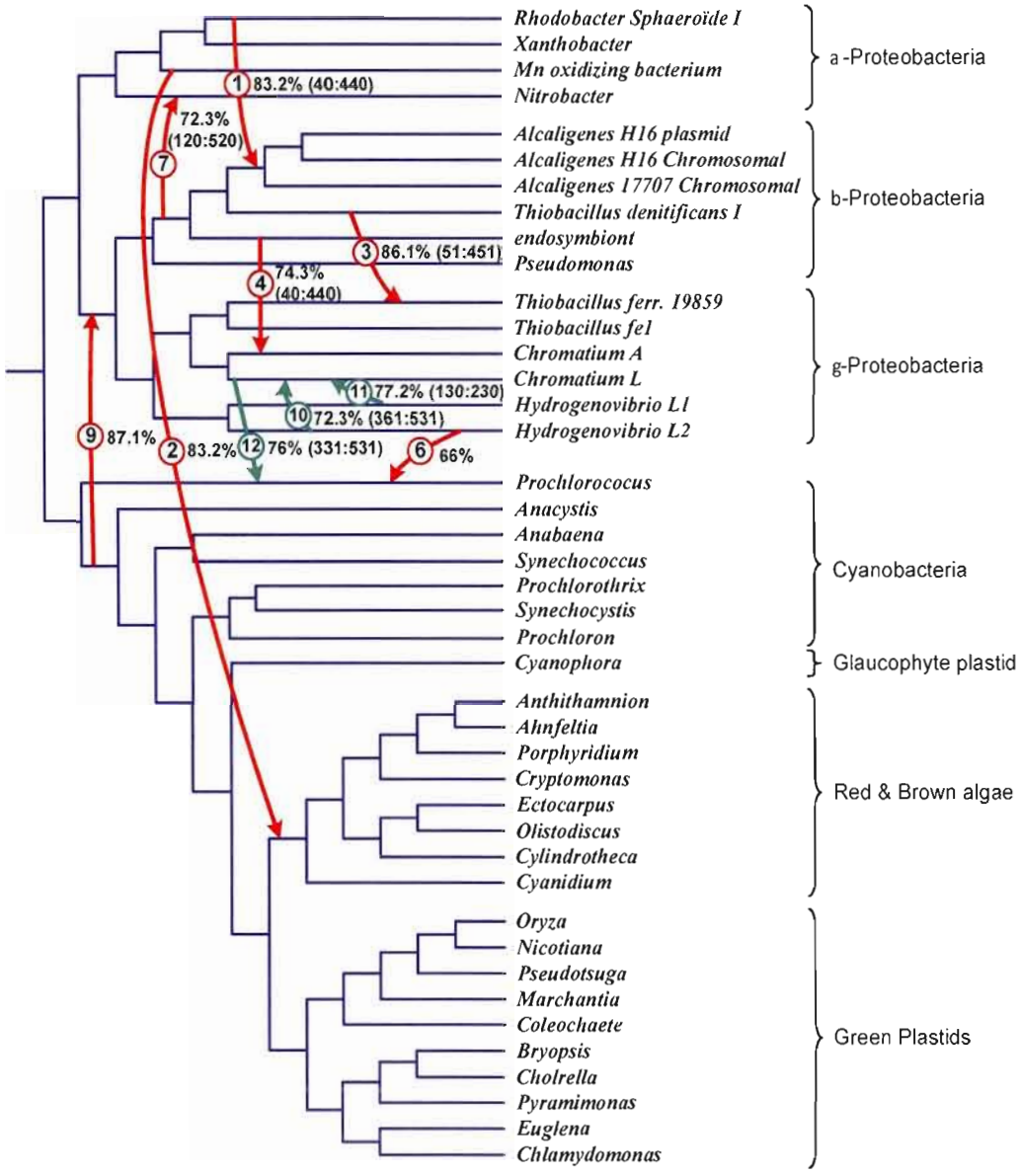


Figure 4.11 Les transferts partiels du gène *rbcL* obtenus en appliquant le deuxième algorithme de détection de THG partiels ; 10 THG partiels ont été détectés. Les transferts analogues à ceux détectés sur la figure (4.10 – HGT complets) ont les mêmes numéros. Les transferts absents sur la figure 4.10 sont numérotés de 10 à 12. Le score de bootstrap des transferts partiels et des intervalles affectés sont indiqués. Pour les transferts 2, 6 et 9 (affectant tout l'ASM) l'intervalle n'est pas indiqué.

4.4.3.2 Détection des transferts partiels du gène *mutU*

Nous avons aussi examiné l'évolution du gène *mutU* (MMR-mismatch repair) d'*Escherichia Coli*, originalement discutée dans Denamur *et al.* (2000). Le *mismatch repair* désigne le système de reconnaissance et de réparation des mésappariements de l'ADN. Ce mécanisme est conservé dans l'évolution, depuis les bactéries, jusqu'à l'homme. Il est essentiel pour maintenir l'intégrité de l'information génétique contenue dans le génome au cours des multiples divisions cellulaires. Denamur et ses collègues ont exploré la possibilité que la carence en MMR émergeant dans la nature ait laissé quelques empreintes dans les génomes bactériens et ont montré que, quand ils sont comparés à des gènes de maintenance des fonctions de base de la cellule, les gènes MMR sont fortement composés de séquences dérivant de différentes lignées phylogénétiques. Les gènes MMR de *E. coli*, *mutS*, *mutL*, *mutH* et *mutU* (*uvrD*), et deux gènes de contrôle, *mutT* et *recD*, ont été partiellement séquencés à partir de 30 souches différentes pour tester l'hypothèse de transfert horizontal. Denamur *et al.* (2000) ont comparé les phylogénies de gènes obtenues à l'arbre de référence (i.e., arbre d'espèces) reconstruit à partir des génomes complets et ont trouvé plusieurs conflits topologiques, allant d'un conflit unique (pour *mutT*) à plusieurs conflits importants (pour *mutS*). Pour tester si les conflits topologiques étaient dus à des THG ou à des artéfacts de reconstruction, les auteurs ont utilisé la méthode de l'*Incongruence Length Difference* (ILD, Farris *et al.*, 1994) et ont conclus que les incongruences étaient significatives et provenaient de transferts horizontaux de gènes. La figure 4.13 montre les transferts partiels hypothétiques du gène *mutU* dans l'arbre d'évolution des souches de *E. Coli* trouvés par Denamur *et al.* (2000). En raison d'un niveau de mosaïcité très élevé des gènes MMR, la souche ECOR 37 n'a pas de position phylogénétique claire dans l'arbre des souches d'*E. Coli* (voir la figure 4.12, tirée de Denamur *et al.*, 2000, où cette souche n'a pas été incluse dans l'ensemble des feuilles de l'arbre).

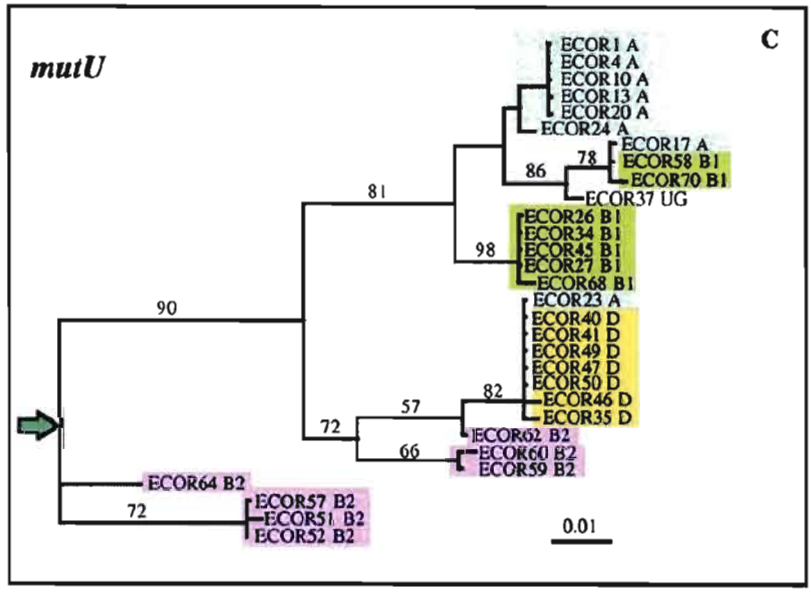


Figure 4.12 La phylogénie du gène *mutU* (Denamur *et al.*, 2000).

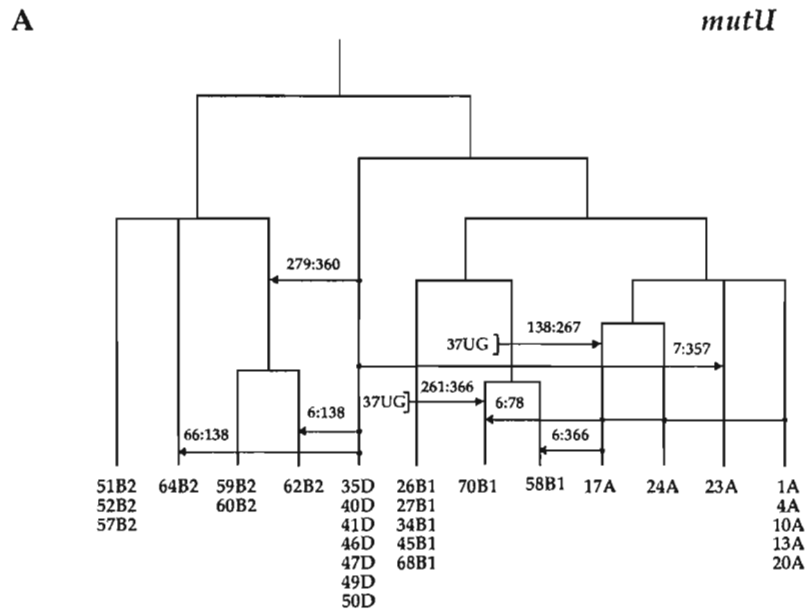


Figure 4.13 Les transferts partiels du gène *mutU* retrouvés par Denamur *et al.* (2000).

Notre deuxième algorithme (utilisant la fenêtre coulissante) de détection des transferts partiels a été appliqué sur les ASM du gène MMR *mutU*, en utilisant trois différentes tailles de fenêtres coulissantes : 100, 150 et 200 nucléotides avec un pas de progression de 10 sites. La taille totale de l'ASM de *mutU* était de 384 nucléotides. Pour reconstruire l'arbre de *mutU*, nous avons utilisé le modèle de substitutions HKY85 (Hasegawa *et al.*, 1985) et les paramètres par défaut de *PHYML*. En raison de la forte similarité entre les séquences d'ADN, de multiples arbres de gène partiels non résolus ont été trouvés. Tous les arbres partiels, ayant un bootstrap moyen inférieur à 50%, ont été exclus de l'analyse (i.e., non traités par l'algorithme *HGT-Detection*).

La figure 4.14 présente les 8 transferts les plus significatifs inférés par notre algorithme (les transferts ayant un support de bootstrap supérieur à 40% sont représentés). Pour chaque transfert, la direction, les espèces impliquées, le support de bootstrap et les intervalles associés à l'ASM originale sont représentés.

Par exemple, les THG 1, 3 et 4, avec un support de bootstrap de 60%, 65% et 46%, respectivement, correspondent à trois transferts similaires trouvés par Denamur *et al.* (2000, voir la figure 4.13). Un analogue exact du THG 4 n'a pas été retrouvé, mais un transfert très proche a été inféré. Le THG 2 détecté par notre algorithme a aussi été identifié par Denamur *et al.* (2000), mais il va dans la direction opposée. Il est à noter que tous les 8 transferts trouvés par Denamur *et al.* (2000) ont aussi été prédits par notre algorithme, mais quatre d'entre eux n'ont pas été représentés dans la figure 4.14 en raison de leur faible support de bootstrap. Nous avons aussi identifié quatre 4 nouveaux transferts partiels (les THG 5, 6, 7 et 8) avec de hauts scores de bootstrap (respectivement, 63%, 94%, 75% et 70%). Mentionnons que la solution générée par l'algorithme *HGT-Detection* pour l'inférence des transferts complets (Boc *et al.*, 2010) inclut seulement les THG 2 et 3 de la figure 4.14. Tous les autres THG ont été différents et avaient de faibles scores de bootstrap.

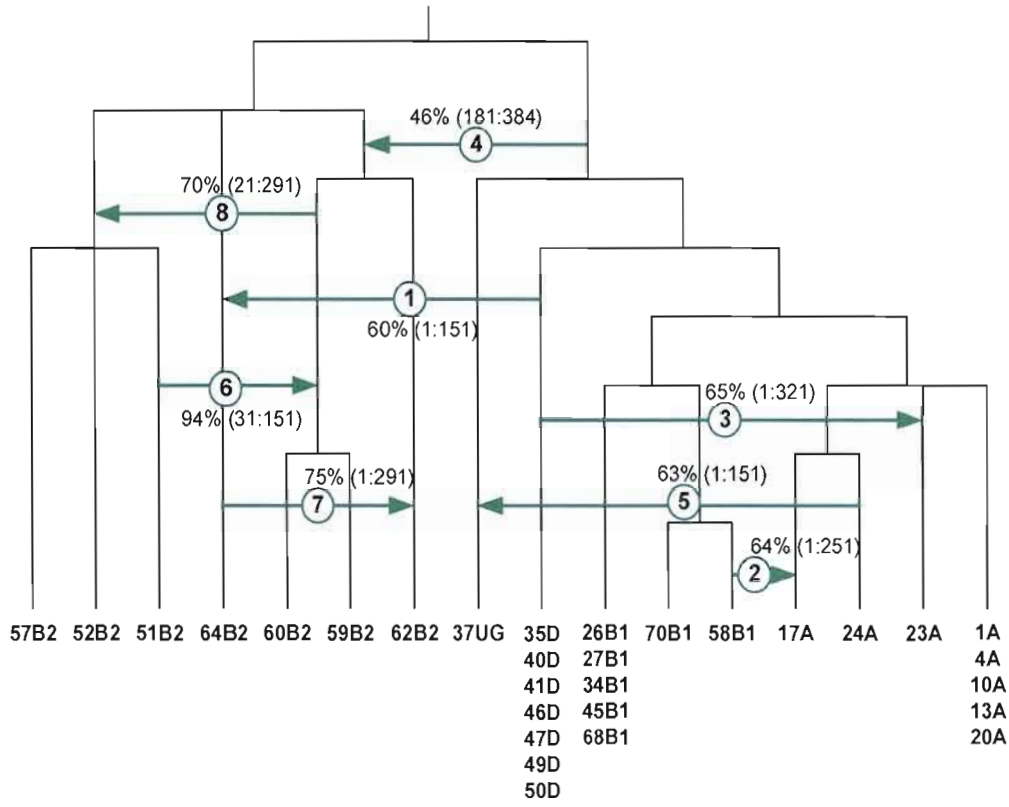


Figure 4.14 Les transferts partiels du gène *mutU* trouvés par le deuxième algorithme de détection des THG partiels. Les score de bootstrap et les intervalles de l'ASM affectés sont indiqués à côté de chaque transfert.

4.5 Discussion et conclusion

Nous avons décrit deux algorithmes pour la prédiction des transferts horizontaux de gènes partiels suivis d'une recombinaison intragénique. Ces algorithmes permettent d'identifier les origines des gènes mosaïques. Au meilleur de notre connaissance, ce problème pertinent de biologie computationnelle n'a pas été convenablement traité dans la littérature (par exemple, les méthodes de Denamur *et al.*, 2000 et Makarenkov *et al.*, 2006b n'incluent aucune validation des transferts obtenus ou de simulations nécessaires pour tester les performances des méthodes).

Le premier algorithme proposé se base sur le critère des moindres carrés, alors que le deuxième utilise une procédure de fenêtre coulissante qui, par conséquent, analyse les fragments de l'alignement de séquences multiples. La taille de la fenêtre coulissante doit être

ajustée en fonction des informations existantes sur les gènes et les espèces étudiés. L'utilisation de petites tailles pour la fenêtre coulissante peut permettre de détecter de plus petits transferts partiels avec une meilleure efficacité (i.e., des THG affectant de petits intervalles de l'ASM donné), mais cela augmente aussi le temps d'exécution du deuxième algorithme. Pour chaque position de la fenêtre, un arbre partiel est inféré et un scénario de transferts est calculé, en réconciliant l'arbre partiel de gène obtenu et l'arbre d'espèces donné. Une procédure de bootstrap, permettant d'évaluer le support de bootstrap de tous les THG partiels et prenant en compte l'incertitude des arbres de gène partiels a aussi été développée.

Les exemples considérés dans la section d'applications suggèrent que le deuxième algorithme peut être utile pour confirmer ou exclure les transferts complets inférés en utilisant n'importe quel algorithme de détection des THG.

Notre étude de l'évolution du gène *rbcL* pour 42 espèces de protéobactéries, cyanobactéries et plastides (Delwiche et Palmer, 1996) et celle de l'évolution du gène de MMR *mutU* pour 30 souches de *E. Coli* (Denamur *et al.*, 2000) ont montré que la plupart des transferts identifiés (six sur huit pour chaque jeu de données) pourraient être, en fait, des transferts partiels suivis de la recombinaison intergénique.

Les simulations Monté-Carlo effectuées pour le deuxième algorithme montrent qu'il peut être efficace dans plusieurs situations pratiques. Alors, le taux de détection moyen (calculé pour des arbres avec 8, 16, 32 et 64 feuilles et pour 1 à 5 transferts générés permettant le chevauchement des fragments de séquences affectées par les THG) était environ 80%, tandis que le taux moyen de faux positifs était environ 30%.

L'information sur les THG partiels et leur score de bootstrap peut être incorporée dans un modèle d'évolution étendu qui prend en compte le transfert horizontal de gènes, la duplication ancestrale et la perte de gènes (e.g., l'incongruence topologique donnant lieu à des transferts complets et/ou partiels avec un faible support de bootstrap peut, en effet, être due à la duplication du gène suivie de sa perte chez certaines espèces). Les algorithmes proposés peuvent être aussi appliqués à une échelle génomique pour estimer la proportion de gènes mosaïques dans chaque génome étudié et les taux de transferts complets et partiels entre les espèces impliquées. Plusieurs statistiques intéressantes concernant la position et la

fonctionnalité des fragments génétiques affectés par les transferts horizontaux, aussi bien que le taux de transferts inter- et intra-espèces, peuvent être calculées en utilisant les techniques discutées. Comme n'importe quelle méthode d'analyse phylogénétique, les algorithmes présentés sont sujet à quelques artéfacts. Les principaux d'entre eux sont l'attraction des longues arêtes, les taux d'évolution inégaux et les situations quand les transferts coïncident, ou presque, avec des évènements de spéciation. Dans le futur, il sera important d'investiguer l'impact de ces artéfacts sur l'identification des transferts horizontaux de gènes partiels.

[Cette page a été laissée intentionnellement blanche]

CHAPITRE V

APPLICATION DE L'ALGORITHME DE DÉTECTION DES TRANSFERTS HORIZONTAUX À L'ETUDE DE L'ÉVOLUTION DES LANGUES INDO-EUROPÉENNES

5.1 Introduction

De nombreux parallèles entre le processus d'évolution en linguistique et l'évolution biologique selon Darwin ont été observés. Atkinson et Gray (2005) ont d'ailleurs présenté un tableau des parallèles conceptuels les plus importants caractérisant l'évolution biologique et linguistique. D'importantes études ont considéré des méthodes phylogénétiques et leurs applications aux données linguistiques (Gray et Atkinson, 2003; Rexová *et al.*, 2003; Atkinson et Gray, 2005). Par exemple, un des domaines largement étudié a été l'évolution des langues Indo-Européennes (IE) (Diamond et Bellwood, 2003). Le jeu de données de 87 langues IE collecté par Dyen *et al.* (1997) a été analysé par plusieurs chercheurs et a servi de base pour l'inférence d'arbres (Gray et Atkinson, 2003; Rexova *et al.*, 2003) et de réseaux (Atkinson et Gray, 2005) phylogénétiques représentant les relations d'évolution entre les langues IE. Malheureusement, ni les arbres phylogénétiques, ni les split-graphes (Bandelt et Dress, 1992a, 1992b) ne peuvent être utilisés pour prédire, et, représenter le phénomène d'emprunt de mots (Haugen, 1950 et Cannon, 1999). En effet, la principale supposition pour la représentation en arbre était que l'évolution des langues a été strictement divergente, que chaque langue a été transmise comme un tout et que la fréquence d'emprunts de mots (i.e., transmission horizontale de mots individuels) entre les langues a été faible. D'autre part, une représentation en split-graphe, qui peut être obtenu en utilisant l'algorithme de décomposition en splits (Bandelt et Dress, 1992) ou l'algorithme *NeighborNet* (Bryant et Moulton, 2004), affiche les plus évidents, mais toujours implicites, signaux conflictuels existant dans les

données linguistiques ou phylogénétiques. Par exemple, les split-graphes peuvent être utilisés pour visualiser les caractéristiques hybrides de la langue Créole (Atkinson et Gray, 2005 et Bryant *et al.*, 2005), mais ne peuvent pas dresser explicitement un portait des évènements d'emprunts de mots (WBE, *word borrowing events*) qui sont équivalents à des transferts horizontaux de gènes (THG) en biologie.

Le THG est un des mécanismes majeurs contribuant à la diversification des génomes microbiens (Doolittle, 1999, Gogarten *et al.*, 2002 et Koonin, 2003). Dans une étude récente, Greenhill *et al.*, (2009) ont vérifié si la transmission horizontale invalidait l'utilisation des méthodes phylogénétiques en simulant des emprunts de mots entre les langues naturelles. Ils ont conclu qu'un niveau d'emprunts réaliste entre les cultures n'invalidait pas l'utilisation des arbres phylogénétiques en évolution linguistique. Cependant, l'étude de Greenhill *et al.* (2009) n'a pas considéré séparément l'évolution de chaque mot. Notons que les évolutions de mots individuels peuvent être très différentes. L'arbre d'évolution des langues résume l'histoire de mots individuels, mais souligne seulement la tendance verticale de l'évolution des langues bien que, pour certaines langues, le niveau d'emprunts puisse être très significatif. Par exemple, l'anglais est une langue germanique, mais elle a emprunté environ 50% de son lexique du français et du latin (Pagel, 2000).

5.2 Application de l'algorithme de détection des THG pour modéliser les emprunts de mots entre les langues Indo-Européennes

5.2.1 Description des données

La base de données organisée par Dyen *et al.* (1997) inclut 200 mots de la liste Swadesh (1952). La liste Swadesh est une des diverses listes de mots de signification basique, constituée par M. Swadesh dans les années 1940-50, qui est couramment utilisée en lexicostatistique (évaluation quantitative de la parenté des langues) et en glottochronologie (datation des divergences entre les langues). Pour chacun des 200 mots de la liste Swadesh, la base de données contient les traductions utilisées dans 95 variétés de langues (87 d'entre elles ont été utilisées dans cette analyse) regroupées par Dyen *et al.* (1997) en ensembles de cognats (i.e., groupes de mots apparentés ayant une racine commune). Le groupement par parenté a été faite seulement pour les traductions ayant le même sens (Dyen *et al.*, 1997). Deux traductions, dans deux langues différentes, étaient identifiées comme apparentées (i.e.,

cognats) si pour ces deux langues, elles avaient une histoire évolutive ininterrompue depuis une forme ancestrale commune. Les traductions connues pour être reliées par des emprunts ou par des similarités accidentelles étaient mises dans une classe séparée. Dans un petit nombre de cas, il était difficile de différencier les cognats des emprunts ou des similarités accidentelles ; de telles traductions ont été classifiées comme cognats douteux. Cette base de données a été utilisée par Gray et Atkinson (2003) pour inférer l'arbre d'évolution des langues IE.

Nous avons modifié quelques ensembles originaux de cognats en leur ajoutant les mots empruntés connexes. Dans certains cas, des mauvais cognats ont été supprimés. Aussi, quand c'était approprié, de nouveaux ensembles, composés de cognats douteux, ont été créés (selon Dyen *et al.*, 1997). La base de données modifiée est disponible à l'adresse URL suivante: http://www.info2.uqam.ca/~makarenv/BL/index.html.

Dans notre étude, nous avons aussi subdivisé les 200 mots de la liste Swadesh (1952) en deux catégories : lexicale (incluant les noms et les verbes, 138 mots au total) et fonctionnelle (incluant les adjectifs, les pronoms, les conjonctions et les déterminants, 62 mots au total), et deux analyses indépendantes, une pour chaque catégorie, ont été effectuées.

5.2.2 Méthodologie

L'algorithme de détection des THG complets (voir chapitre III et Boc *et al.*, 2010) a été appliqué ici dans un contexte biolinguistique pour inférer un réseau phylogénétique des langues IE. Un nombre important de nouvelles caractéristiques a été ajouté à l'algorithme de base pour le rendre applicable à l'identification des emprunts de mots. Quand cet algorithme est exécuté dans un contexte biologique, il identifie les THG d'un gène donné pour un ensemble d'espèces considérées, réconciliant ainsi les arbres phylogénétiques d'espèces et de gène. À chaque étape du processus de réconciliation, un ensemble de THG compatibles est inféré. En établissant les parallèles entre les processus d'identification des THG et des emprunts de mots, l'arbre des langues IE (voir la figure 4 dans Gray et Atkinson, 2003) joue le rôle de l'arbre d'espèces et l'arbre de mot, représentant l'évolution d'un cognat particulier, joue le rôle de l'arbre de gène. La procédure algorithmique incluait les trois principales étapes décrites ci-dessous :

Étape 1. Soit L l'arbre enraciné de 87 langues IE inféré par Gray et Atkinson (2003). La figure 5.1 illustre la topologie et les principaux groupes de langues de cet arbre.

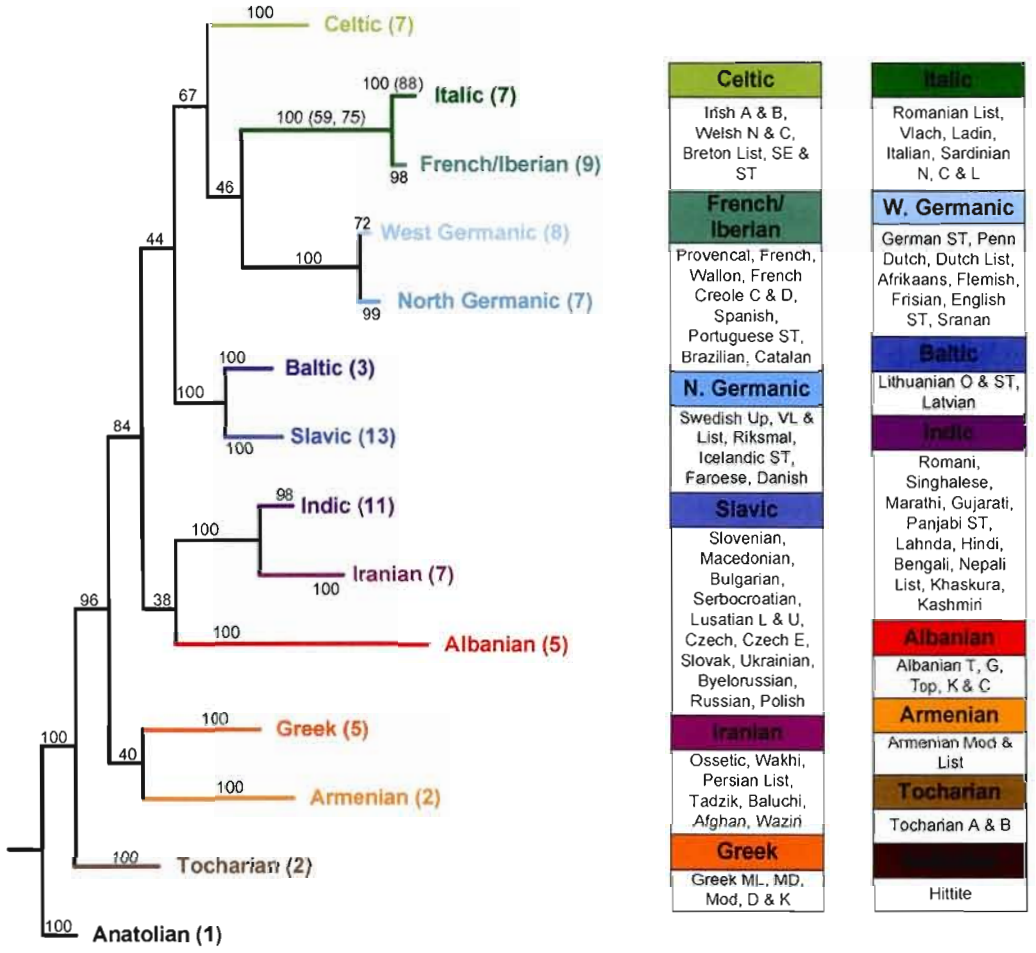


Figure 5.1 L'arbre d'évolution des langues IE pour 14 groupes principaux. Les nombres sur les branches représentent leur score de bootstrap. Le nombre de langues appartenant à chaque groupe est indiqué entre parenthèses.

Nous avons considéré les 200 mots de la liste Swadesh regroupés en 1484 cognats. Pour chaque ensemble de cognats, c, nous avons calculé la matrice de distances $\mathbf{W}_c$, incluant les distances entre les traductions incluses dans c. La distance entre les traductions i et j dans c était calculée à l'aide de la formule suivante :

$$Normalized\_LD = \frac{Modified\_Levenshtein\_distance(i,j)}{length(i) + length(j)} . \quad (1)$$

Cette formule utilise une version modifiée de la distance de Levenshtein (1966) qui est normalisée par les longueurs des traductions *i* et *j*. La distance classique de Levenshtein considère que la distance entre deux lettres différentes dans les traductions alignées est 1. Comme cette règle ne prend pas en compte les ressemblances entre certaines lettres, les distances entre les lettres étroitement reliées ont été introduites : la distance de 0,5 entre D et T, F et V, I et Y, C et Q, Q et K, K et C, S et Z, V et W ; la distance de 0,25 était assignée si la différence entre deux mots était causée par un apostrophe ou un tiret. De plus, toutes les lettres dupliquées dans un mot étaient remplacées par une seule et une valeur de 0,25 était ajoutée à la distance de Levenshtein pour prendre en compte ce remplacement.

Pour chaque matrice de distances $\mathbf{W}_c$, nous avons inféré un arbre phylogénétique non-enraciné *W_c* en utilisant l'algorithme NJ (Saitou et Nei, 1987). La figure 5.2 montre la distance topologique de Robinson et Foulds (Robinson et Foulds, 1981), normalisée par sa valeur maximale de 2*n*-6 (pour deux arbres binaires avec *n* feuilles) entre chacun des 1484 arbres de mots *W_c* et la version réduite de l'arbre de langues *L*, notée *L_c*.

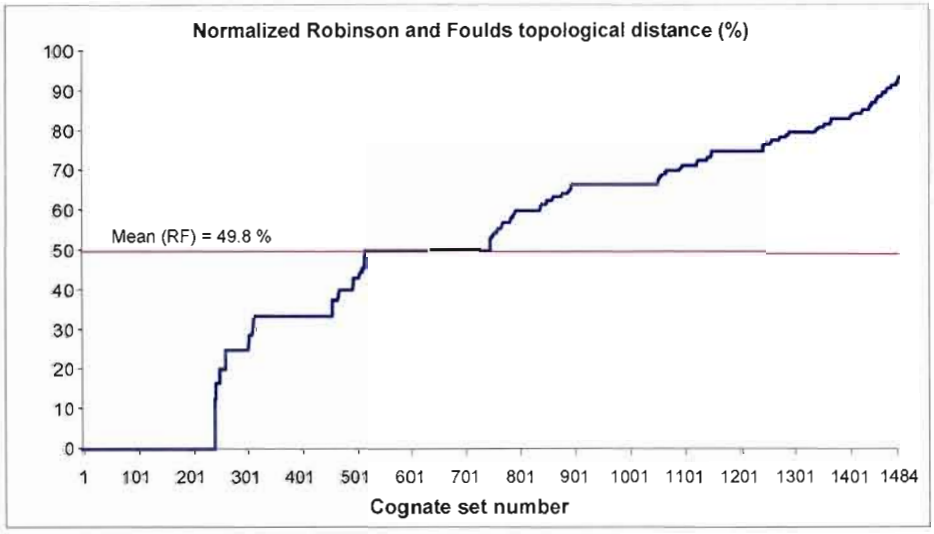


Figure 5.2 La distance topologique de Robinson et Foulds normalisée entre chaque arbre de mot (1484 au total) et l'arbre de langues réduit correspondant.

L'arbre de langues réduit *L_c*, correspondant à l'ensemble de cognats *c*, a été obtenu en supprimant d'abord toutes les arêtes de *L* relatives aux langues absentes dans *c* et puis en supprimant tous les nœuds de degré 2. Donc, les arbres *W_c* et *L_c* avaient le même nombre de feuilles rendant possible le calcul de la distance RF et l'exécution de l'algorithme de détection des THG. La valeur moyenne obtenue de la distance normalisée RF est de 49,8% (voir la ligne rouge droite sur la figure 5.2). Une telle valeur moyenne suggère un important niveau de contradiction entre l'arbre de langues *L* et les arbres de mots *W_c* ($c = 1, \ldots, 1484$). Un arbre de langues *L_c* pouvait contenir des multifurcations (i.e., des nœuds internes de degré supérieur à 3) dans le cas où une ou plusieurs langues avaient de multiples traductions incluses dans l'ensemble de cognats *c*. Dans le cas de traductions identiques (e.g., la traduction *frukt* est la même pour le biélorusse, le polonais, le russe et l'ukrainien sur la figure 5.4), la topologie du clade comprenant ces traductions dans l'arbre de mots était organisée de façon identique à la topologie du clade correspondant dans l'arbre de langues.

Étape 2. Nous avons appliqué l'algorithme de détection de THG (Boc *et al.*, 2010) pour inférer les emprunts de mots, en considérant l'arbre de langues réduit *L_c* comme l'arbre d'espèces et l'arbre de mot *W_c* comme l'arbre de gène. L'algorithme a été exécuté 1484 fois et 1484 scénarios d'emprunts de mots ont été identifiés. L'algorithme appliqué reposait sur le critère de dissimilarité de bipartitions et recherchait un scénario de coût minimal d'opérations SPR (Subtree Prune and Regraft) qui sont nécessaires pour transformer l'arbre de langues et l'arbre du mot considéré. Une version spécifique de l'algorithme, permettant d'utiliser des arbres de mots non-enracinés a été développée. L'arbre de langues était toujours enraciné comme le montre la figure 5.1. Comme la base de données de Dyen (1997) ne comprend aucune traduction pour les langues hittite et tokhariennes A et B, dépendant, respectivement, des groupes Anatolien et Tokharien, ces langues n'ont pas été considérées dans notre analyse. Aussi, les contraintes suivantes ont été adoptées dans cette étude pour appliquer l'algorithme de détection dans le contexte biolinguistique. Comme un mot emprunté pouvait changer en cours d'évolution et dans le but de donner la priorité aux groupes de langues IE originales, nous avons rendu certains clades « incassables ». En particulier, la suppression, par une opération SPR associée à un évènement d'emprunt de feuilles ou de sous-arbres d'un clade d'arbre de langues, était interdite si la distance moyenne non-normalisée de Levenshtein entre

les traductions incluses dans ce clade était plus petite que 1,5 (cette valeur a été déterminée expérimentalement). Cependant, un clade entier pouvait être déplacé vers une autre partie de l'arbre de langue, quelle que soit la distance moyenne normalisée entre ses traductions. Nous avons aussi exclu tous les échanges entre deux clades si la distance normalisée de Levenshtein entre les traductions incluses dans ces clades était plus grande que 0,35 (cette valeur a aussi été déterminée expérimentalement).

Étape 3. À cette étape, nous avons combiné tous les résultats obtenus provenant des scénarios d'emprunts de mots pour calculer des statistiques (voir les figures 5.5, 5.6 et 5.7). Si plusieurs groupes de langues étaient impliqués dans l'emprunt d'un mot, les transferts obtenus étaient pondérés prenant en compte de toutes les traductions impliquées. La figure 5.3a illustre cette situation : le transfert entre le clade incluant des traductions provenant des groupes G1 et G4 et le clade incluant des traductions provenant des groupes G2 et G3 pourrait être pris en compte comme suit :

1/2 WBE pour G1 -> G3; 3/2 WBE pour G1 -> G2,

1/2 WBE pour G4 -> G3; 3/2 WBE pour G4 -> G2.

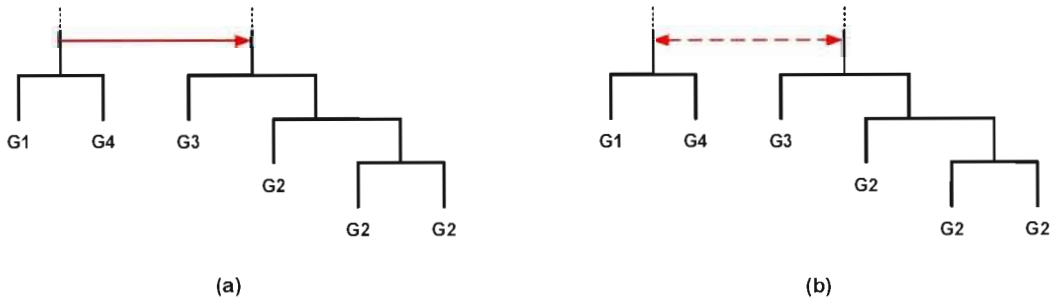


Figure 5.3 Deux exemples d'emprunts impliquant plusieurs groupes de langues.

Les évènements d'emprunts représentés sur la figure 5.3a seraient décomposés en quatre transferts, où les poids dépendraient du nombre de traductions présentes pour chaque groupe. Le transfert montré sur la figure 5.3b représente une situation où sa direction est incertaine (il est représenté par une double flèche). Dans ce cas, chacun des deux transferts comptera pour la moitié d'un transfert complet.

Nous avons donc évalué, le nombre des WBE pour les 12 groupes principaux de langues IE, puis les pourcentages de mots affectés par un emprunt dans chaque groupe. Les taux de mots empruntés entrants (figure 5.5c) et sortants (figure 5.5d) ont été calculés. Ces calculs ont été tout d'abord effectués pour tous les 200 mots de la liste Swadesh (figure 5.5), puis séparément, pour les mots des catégories lexicale (figure 5.6) et fonctionnelle (figure 5.7).

5.3 Résultats et discussion

Au total, 1484 ensembles de cognats comprenant au moins 4 traductions ont été analysés. La valeur de 49,8% de la distance topologique moyenne de Robinson et Foulds entre les arbres de langues réduits et les arbres de mots (un arbre de mots par cognat a été inféré, voir la figure 5.2) suggère une importante contradiction entre l'évolution verticale des langues et l'histoire individuelle de chaque mot. L'algorithme de détection des THG (Boc *et al.*, 2010) a été appliqué pour prédire les évènements d'emprunts de mots caractérisant l'évolution de chaque cognat et les statistiques correspondantes ont été calculées.

Il est à noter que certaines des similarités de traduction, même celles entre les traductions dépendantes de différents groupes de langues, peuvent être dues à des ressemblances accidentelles. Par exemple, le mot anglais *bad* n'a pas de cognats dans d'autres langues. Le farsi a le mot *bad* dans plus ou moins le même sens que l'anglais, mais ceci est plutôt considéré comme une coïncidence par les étymologistes (voir *Online Etymology Dictionary*). Les formes de mots divergent quand elles sont retracées dans le temps (le mot *bad* du farci vient de *vat* en persan), mais les convergences accidentelles existent également à travers plusieurs langues. Cependant, de telles ressemblances fortuites entre les traductions ayant la même signification sont beaucoup moins fréquentes que les ressemblances dues à l'emprunt de mots.

L'évolution présumée du mot *fruit* est présenté à la figure 5.4. L'évolution de ce mot, ayant comme origine le mot Proto-Indo-Européen *bhrug* (voir *Online Etymology Dictionary*), englobe 7 emprunts hypothétiques (représentés par les lignes en pointillé dans la figure 2). Le mot en vieux français *fruit* est dérivé du latin *fructus* (*Online Etymology Dictionary* et *Webster's Third New International Dictionary*). En plus du transfert du vieux français,

signifiant "jouissance, produit, bénéfice, production, revenu" au moyen anglais signifiant "fruits et légumes", ce mot a aussi été emprunté par le russe depuis l'allemand ou le hollandais (Chernih, 2001), possiblement à travers le polonais (Fasmer, 1950-1958), au début du 17ème siècle sous le règne de Pierre le Grand. Par exemple en anglais, le mot *fruit* a remplacé le mot natif du moyen anglais *ovet* "fruit" (provenant du vieil anglais *ofett*), du moyen anglais *wastun* et *wastom* "fruit, croissance" (provenant du vieil anglais *waestm*), et du moyen anglais *blede* "fruit, fleur, progéniture" (provenant du vieil anglais *blēd*) (*Online Etymology Dictionary* et *Webster's Third New International Dictionary*). Les cinq autres emprunts probables de mots incluent les transferts de l'italien vers le grec moderne (*Greek etymological dictionary*), l'albanais, l'espagnol et le portugais, du vieux français vers l'irlandais et le breton, et d'une des langues germaniques vers le provençal.

Les emprunts de mots les plus fréquents identifiés par une version spécifique de notre algorithme de détection des THG ont été ajoutés à l'arbre de langues pour représenter les échanges de mots les plus importants qui se sont produits durant l'évolution des langues IE (figure 5.5a). Si la proximité géographique peut expliquer la plupart des échanges fréquents (e.g., entre les groupes Germaniques du nord et de l'ouest, ou entre le Français/Iberian et l'Italique), certains parmi eux arrivent entre des groupes éloignés (e.g., entre les groupes Grec et Baltique, ou Indic et Arménien).

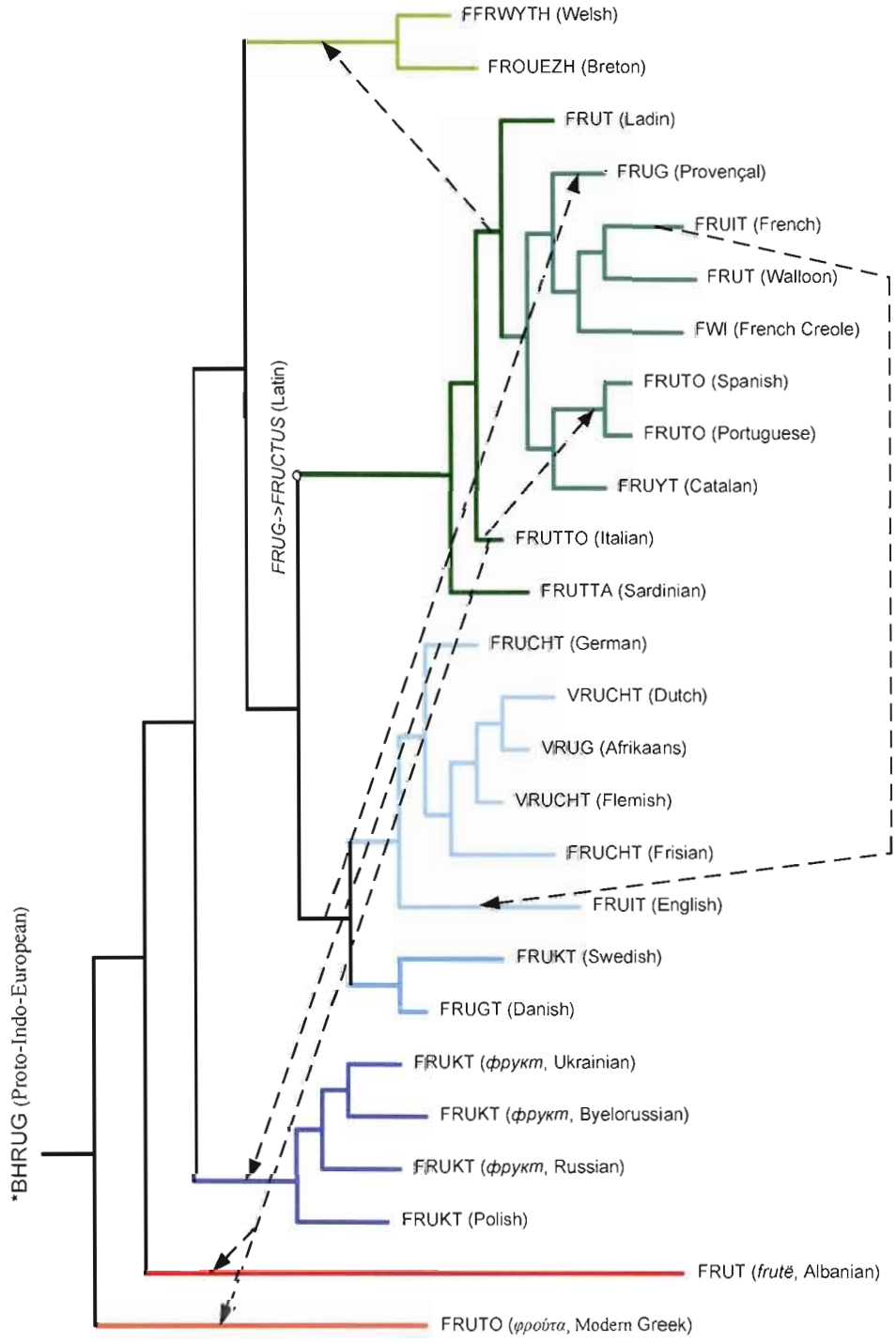


Figure 5.4 Évolution présumée du mot FRUIT ; 7 emprunts de mots hypothétiques sont illustrés par des flèches en pointillé.

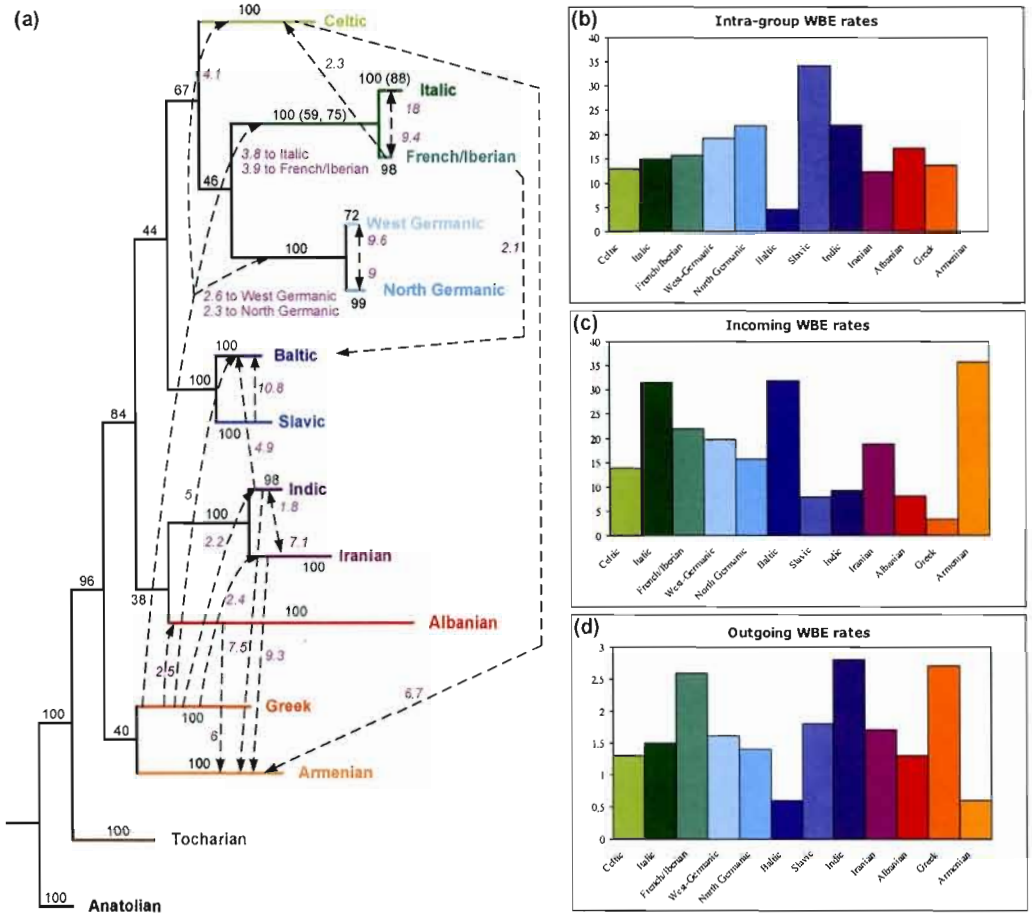


Figure 5.5 Résultats obtenus pour les mots des catégories lexicale et fonctionnelle (i.e., le total des mots). Par exemple, 18% des mots du groupe Italic ont une origine French/Iberian. Les diagrammes sur la droite montre : (b) le taux d'emprunts à l'intérieur des groupes, (c) le taux d'emprunts pour des mots provenant d'autres groupes et (d) le taux d'emprunts pour des mots sortant vers d'autres groupes.

Notons que les similarités entre les langues baltiques et l'ancien grec ont été soulignées depuis longtemps par F. Bopp (1845-56). Par exemple l'arménien, qui forme une branche indépendante de la famille des langues IE, est plus proche du Grec, mais a beaucoup de mots empruntés des langues Indo-Iraniennes comme le pachto et le perse (Comrie, 1981) (voir les valeurs de 9,3% et 7,5% pour les transferts des groupes Iranien et Indien vers l'arménien sur la figure 5.5a). Très tôt, durant la période de sa classification, l'arménien a été même

considéré comme une langue du groupe Iranien en raison de son large nombre de mots empruntés (Comrie, 1981). Notre analyse montre que l'arménien a été aussi influencé par les langues des groupes Albanais et Celtique (figure 5.5a).

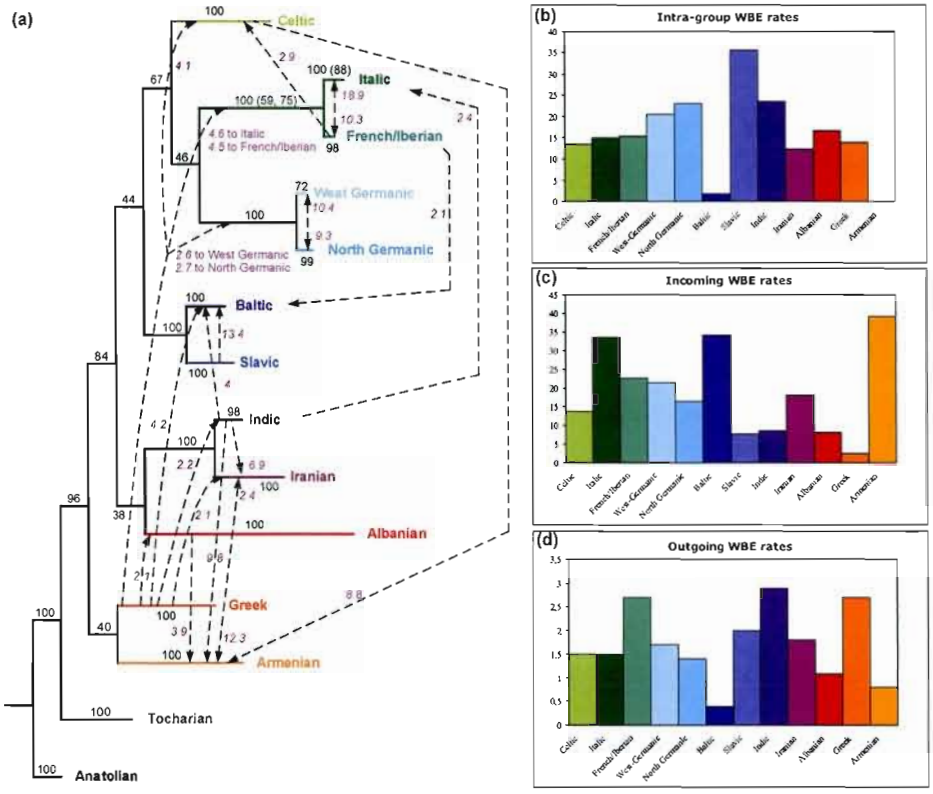


Figure 5.6 Résultats obtenus pour les mots de la catégorie lexicale (les mêmes notations que celles utilisées sur la figure 5.5 sont adoptées ici).

Globalement, 35,4% des traductions considérées ont été affectées par des emprunts, incluant 15,5% découlant de groupes distincts. Les résultats analogues ont été obtenus pour les mots des catégories lexicale (36,3% - noms et verbes) et fonctionnelle (33,3% - adjectifs, pronoms, conjonctions et déterminants). Ces résultats révèlent que le taux d'emprunts de mots ne dépend pas de la catégorie du mot. Les emprunts plus fréquents entre les 12 groupes de langues IE trouvés indépendamment pour les catégories fonctionnelle et lexicale sont illustrés sur les figures 5.6 et 5.7, respectivement. Les résultats détaillés pour chaque ensemble de cognats considéré sont disponibles à l'adresse URL suivante :

http://www.info2.uqam.ca/~makarenv/BL/index.html. Chaque arbre réduit de langues aussi bien que chaque arbre de mot, représentant l'évolution de l'ensemble de cognats correspondant, peuvent être visualisés avec les scénarios d'emprunts obtenus. Tous les 1484 arbres de mots considérés et les scénarios d'emprunts, obtenus en utilisant les algorithmes *NJ* (Saitou et Nei, 1987) et *HGT-Detection* (Boc et al., 2010), ont été vérifiés et, si nécessaire, corrigés manuellement pour assurer la plausibilité des emprunts prédits.

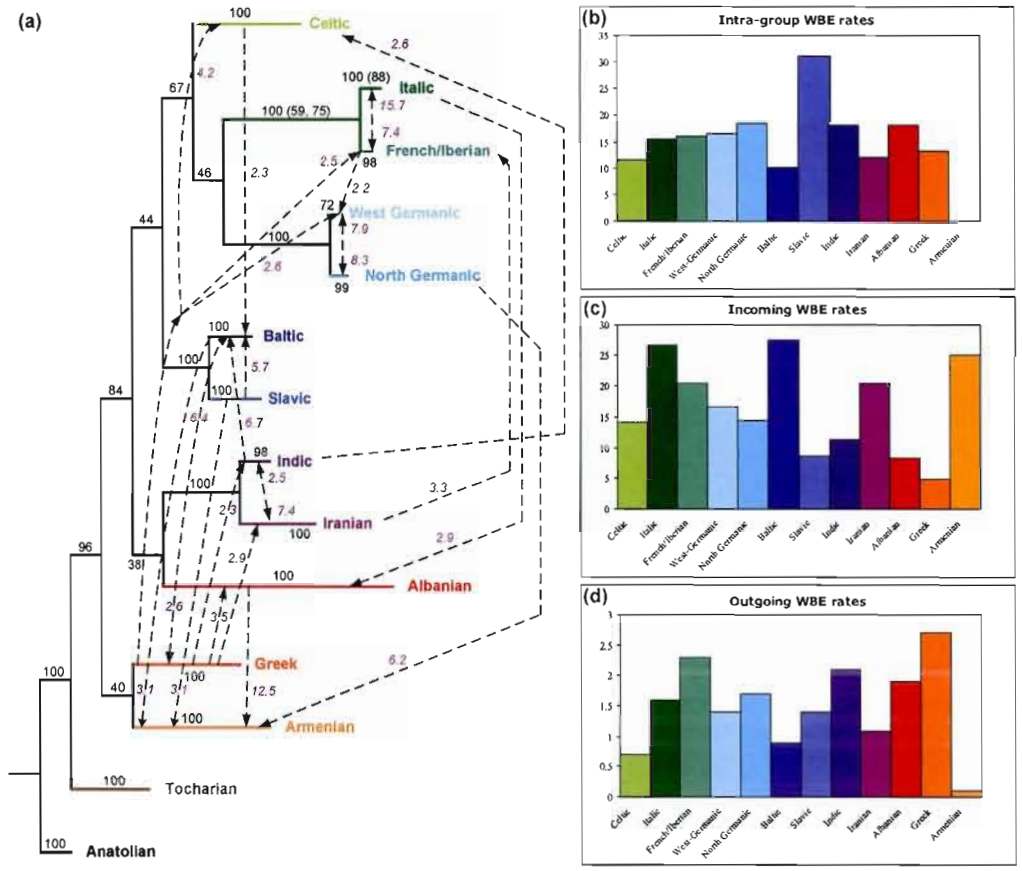


Figure 5.7 Résultats obtenus pour les mots de la catégorie fonctionnelle (les mêmes notations que celles utilisées sur la figure 5.5 sont adoptées ici).

Le diagramme de la figure 5.5 (b, c et d) illustre les taux d'emprunts de mots intra-groupe, entrant et sortant. Les groupes ayant les taux d'emprunts les plus élevés en leur sein sont les groupes Slaves, Germaniques du nord et de l'ouest, Indic et Albanais. Le taux exceptionnel de 34,2% pour des emprunts intra-groupe obtenu pour les langues de la famille Slave démontre les relations proches et des échanges très intensifs entre les nations slaves. Quant aux transferts de mots entrants (figure 5.5c), les trois groupes – Italien, Baltique et Arménien – ont de hauts pourcentages de mots affectés par des échanges provenant de langues de groupes différents. La figure 5.5d suggère que les langues des groupes Indic, Grec et Français/Ibérien ont eu la plus grande influence sur le reste des langues IE. Par exemple, un mot d'Indic a la probabilité de 2,8% d'être un terme emprunté dans un groupe différent.

Alors que retracer l'évolution exacte de chaque ensemble de cognats peut être une tâche très sophistiquée, les emprunts de mots estimés et représentés sur la figure 5.5a* peuvent aider à découvrir les échanges les plus intenses qui se sont produits après la formation des groupes de langues Indo-Européennes et possiblement prédire des évènements relatifs, tels que les guerres, les longues occupations, les migrations ou les importants échanges commerciaux survenus entre les nations impliquées.

---

* Les résultats biolinguistiques présentés dans cette thèse sont décrits dans un article soumis à une revue. Ils diffèrent des résultats présentés dans Boc *et al.* (2010b).

CONCLUSION ET PERSPECTIVES

Dans cette thèse, nous avons présenté trois nouveaux algorithmes pour la détection des transferts horizontaux de gènes ainsi que leurs nombreuses applications. Ces algorithmes exploitent efficacement les différences topologiques ou métriques entre un arbre d'espèces et un arbre de gène pour un même ensemble d'organismes considérés et fournissent une réponse concrète au problème de la modélisation de transferts horizontaux de gènes complets et partiels. Nous avons apporté notre contribution à quatre niveaux, qui sont comme suit :

Première contribution

En considérant le modèle arborescent, où le gène est transféré complètement et supplante entièrement le gène homologue de l'espèce hôte, nous avons développé un nouvel algorithme de détection des THG en nous basant sur l'algorithme décrit dans Makarenkov *et al.* (2006). Nous y avons apporté un grand nombre d'améliorations, notamment sur le plan de la complexité algorithmique, la prise en compte des règles d'évolution et la viabilité des transferts détectés. La première nouveauté majeure consiste en la définition d'une nouvelle mesure d'arbre : la dissimilarité de bipartitions (BD). Cette mesure qui pourrait être vue comme un raffinement de la distance de Robinson et Foulds, s'est montrée plus efficace que tous les autres critères testés (LS, RF et QD) en ce qui concerne la génération d'un scénario de THG optimal pour la réconciliation d'un arbre d'espèces et un arbre de gène. Les simulations Monte-Carlo menées dans le cadre de cette thèse de doctorat ont montré que l'utilisation de la mesure BD était plus appropriée que l'utilisation de LS, RF ou QD quand vient le temps d'effectuer le choix d'un transfert horizontal. La seconde nouveauté majeure est l'ajout d'un processus de validation des transferts détectés. Nous avons alors présenté trois façons de calculer le support de bootstrap d'un transfert dépendamment des données disponibles : (1) les séquences utilisées pour construire les arbres d'espèces et de gène

pouvaient être répliquées ; (2) seules les données de séquences utilisées pour construire l'arbre de gène pouvaient être répliquées ; (3) le bootstrap de THG pourrait être calculé à partir des deux topologies d'arbres seulement. À ce jour, nous sommes les premiers à proposer un tel processus de validation exhaustif et cohérent. Des comparaisons avec les tous meilleurs algorithmes existants, *LatTrans* (Hallett et Lagergren, 2001) et *RIATA-HGT* (Nakhleh *et al.*, 2005; Than et Nakhleh, 2008), ont montré que notre algorithme était au moins équivalent en termes de qualité des résultats, mais surtout plus rapide et donc plus adapté à la détection de transferts horizontaux dans le contexte des génomes complets ou des ensembles de gènes. Cette contribution a donné lieu à trois publications : Makarenkov et al. (2006), Makarenkov *et al.* (2007) et Boc *et al.* (2010a).

Comme n'importe quelle méthode phylogénétique, cet algorithme est sujet à quelques artéfacts. Les principaux sont l'attraction des longues arêtes, les taux inégaux d'évolution et les situations quand les transferts retracés sont temporellement proches des évènements de spéciation. De plus, la différence topologique entre un arbre d'espèces et un arbre de gène peut être due à d'autres mécanismes d'évolution tels que la duplication ancestrale suivie de la perte partielle du gène. Nous ne tenons pas compte de ces évènements évolutionnaires dans notre modèle, bien que de faibles pourcentages de bootstrap de certains transferts puissent aussi les révéler. Il serait important dans l'avenir de mesurer l'impact de ces artéfacts et évènements sur les résultats de détection des transferts horizontaux et de proposer un processus de correction adéquat du modèle original.

Deuxième contribution

Nous avons ensuite présenté une généralisation du modèle de THG complet applicable à un modèle en réseau où l'on considère le transfert partiel d'un gène. Ce modèle, qui implique la formation de gènes mosaïques, est plus complexe car la distance minimale entre deux espèces peut être calculée à travers plusieurs chemins possibles. Deux approches ont été proposées. La première se base sur l'optimisation des distances par les moindres carrés pour retrouver les transferts horizontaux optimaux, ainsi que les portions de gène transférées. Nous avons démontré théoriquement que ce problème d'optimisation est NP-difficile et que les premières simulations menées (non présentées dans cette thèse) ont montré sa viabilité en pratique. La

deuxième approche se base sur une procédure de fenêtre coulissante qui analyse des fragments de l'alignement de séquences. Pour chaque position de la fenêtre, un arbre de gène partiel est inféré et un scénario de THG est calculé en réconciliant l'arbre de gène partiel et l'arbre d'espèces donné. Une procédure de validation, permettant d'évaluer le support de bootstrap de tous les THG partiels possibles et prenant en compte l'incertitude des arbres de gène a aussi été développée. Les deux exemples considérés, ainsi que les simulations menées, suggèrent que la dernière procédure peut être utile pour confirmer ou exclure les transferts complets inférés en utilisant n'importe quel algorithme de détection des THG et qu'elle peut être efficace dans plusieurs situations pratiques. Par exemple, notre étude de l'évolution du gène *rbcL* pour 42 espèces de protéobactéries, cyanobactéries et plastides (Delwiche et Palmer, 1996) a montré que les THG prédits par Delwiche et Palmer pourraient être en fait des transferts partiels suivis de recombinaison intragénique. Ce projet de recherche a donné lieu à trois publications : Makarenkov *et al.* (2006), Makarenkov *et al.* (2008) et Boc *et al.* (2011, soumis).

Un des principaux problèmes rencontrés par cette approche est l'estimation de la taille minimale de la fenêtre. En effet, le modèle d'évolution applicable à l'alignement de séquences multiples (ASM) complètes peut ne pas être applicable à l'ASM localisée dans une fenêtre de taille trop petite. Par conséquent, les différences topologiques entre l'arbre d'espèces et l'arbre de gène partiel peuvent ne pas être dues à des transferts horizontaux. Évidemment, un faible score de bootstrap permettrait de supprimer ces faux positifs. Ce point est un des éléments majeurs qu'il faudra développer à l'avenir en créant, par exemple, une matrice de conversion des distances évolutives entre l'ASM complet et l'ASM partiel. Des simulations complètes doivent être aussi menées pour valider en pratique le modèle de détection des transferts partiels basé sur l'optimisation par les moindres carrés.

Troisième contribution

Finalement, nous avons appliqué une version adaptée de l'algorithme de détection des transferts horizontaux à l'étude de l'évolution des langues Indo-Européennes (IE), en considérant l'arbre des langues IE comme arbre d'espèces et l'arbre de chaque mot étudié comme arbre de gène. Les résultats obtenus nous ont permis de dresser un portrait général

des échanges les plus importants survenus au cours de l'évolution des langues IE et de calculer plusieurs statistiques intéressantes concernant ces échanges. Tout d'abord, nous avons pu établir qu'environ 35,4% des mots considérés ont été affectés par des emprunts, et ce, indépendamment de la catégorie (lexicale ou fonctionnelle) des mots. Puis, nous avons pu corroborer la plupart des échanges trouvés par des études menées par des linguistes. Par exemple, l'emprunt du mot *fruit* par l'anglais depuis le français est bien connu et bien référencé dans la littérature. Cette contribution a donné lieu à une publication : Boc *et al.* (2010b).

La plus grande incertitude de notre approche se retrouve dans la reconstruction des arbres de mots. En effet, on retrouve souvent des ensembles de cognats avec des traductions similaires (ou très similaires) de faibles tailles. L'utilisation de la distance de Levenshtein normalisée, ainsi que des contraintes biolinguistiques, permettent de remédier partiellement à ce problème. Une vérification manuelle des 1484 arbres de mots a aussi été nécessaire. Les travaux futurs seront alors orientés dans le sens de la validation des arbres de mots obtenus depuis des ensembles de cognats. Un des grands défis de l'étude de l'évolution des langues Indo-Européennes est aussi d'arriver à identifier leur origine. Trois hypothèses sont aujourd'hui considérées : (1) l'hypothèse kourganes, la plus admise aujourd'hui ; (2) l'hypothèse anatolienne ; et (3) l'hypothèse de la continuité paléolithique. Nous pensons qu'en affinant notre modèle et en comparant l'ensemble des statistiques obtenues, nous pourrions y apporter de nouveaux éléments.

Quatrième contribution

Nous avons aussi mis en place une plate-forme d'analyse phylogénétique : la version Web de *T-Rex* (Makarenkov, 2001). Initialement décrite en 2001, la version Web de *T-Rex* inclue de nombreuses applications et algorithmes utiles à l'analyse phylogénétique. Les langages de programmation *HTML* et *PHP* ont été utilisés pour le développement du site Web et la gestion des données provenant des formulaires. Des scripts écrits en langage PERL ont été utilisés pour encapsuler l'exécution des programmes externes souvent écrits en langages C/C++. Une base de données biologiques a aussi été mise en place avec le système de gestion de base de données Oracle. Cette base de données permet de récolter des ensembles de

séquences qui pourraient être utilisés directement par les différentes applications incluses dans *T-Rex*.

Nous avons principalement ajoutés à *T-Rex* des algorithmes de reconstruction d'arbres phylogénétiques, de réseaux phylogénétiques et de détection de transferts horizontaux de gènes. Des algorithmes de tracé des arbres (hiérarchique, radiale, axiale) ont aussi été implémentés. L'Annexe A présente une série de copies d'écran du site Web de *T-Rex*. Dans le but d'offrir un environnement plus complet, nous avons ajouté au site des logiciels développés par d'autres chercheurs de renom tels que certains algorithmes du paquet *PHYLIP* (Felsenstein, 1989) ou encore le programme d'alignement de séquences *ClustalW* (Thompson, 1994).

Le site Web de *T-Rex* se trouve en perpétuelle évolution car nous y ajoutons tous les algorithmes développés dans le cadre de ce projet doctoral de même que des projets connexes. Depuis la mise en place officielle de la première version, le site Web www.trex.uqam.ca a reçu plus de 90,000 visites. Les prochaines étapes de développement seront l'ajout d'une interface graphique pour la détection des transferts partiels et pour l'analyse de l'évolution des langues naturelles.

[Cette page a été laissée intentionnellement blanche]

ANNEXE A : EXEMPLES D'INTERFACES WEB DU LOGICIEL *T-REX*

L'annexe A présente quelques exemples d'interfaces de l'algorithme *HGT-Detection* dans la version Web du logiciel *T-Rex* (Tree and Reticulogram Reconstruction) disponible à l'adresse URL suivante : http://www.trex.uqam.ca.

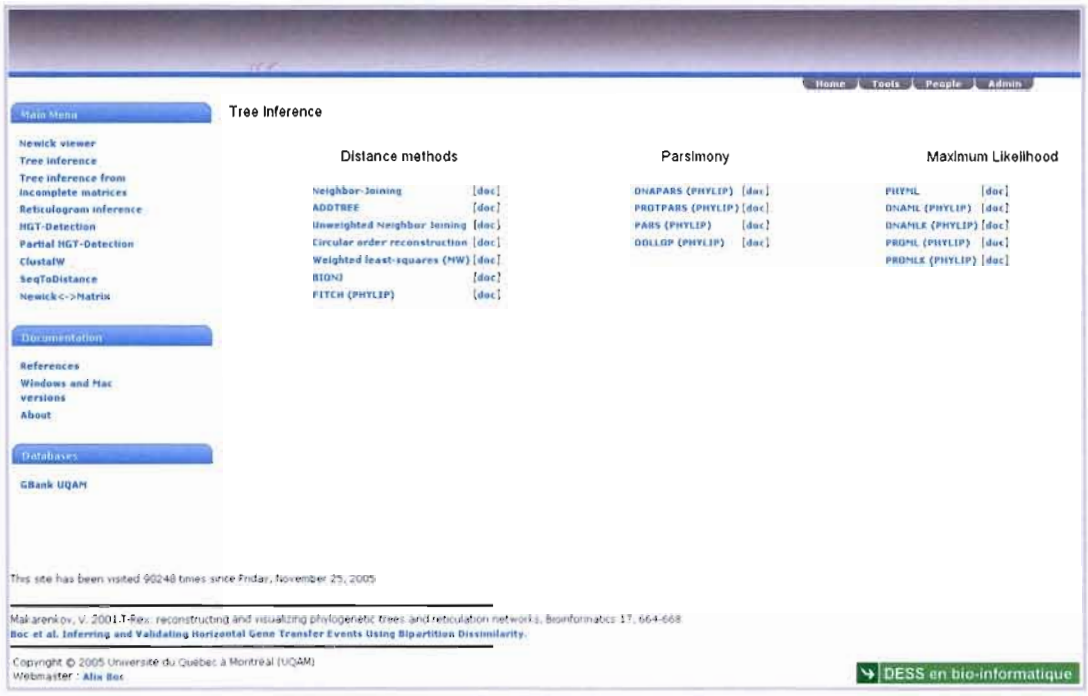


Figure A.1 Interface principale de la version Web de *T-Rex*.

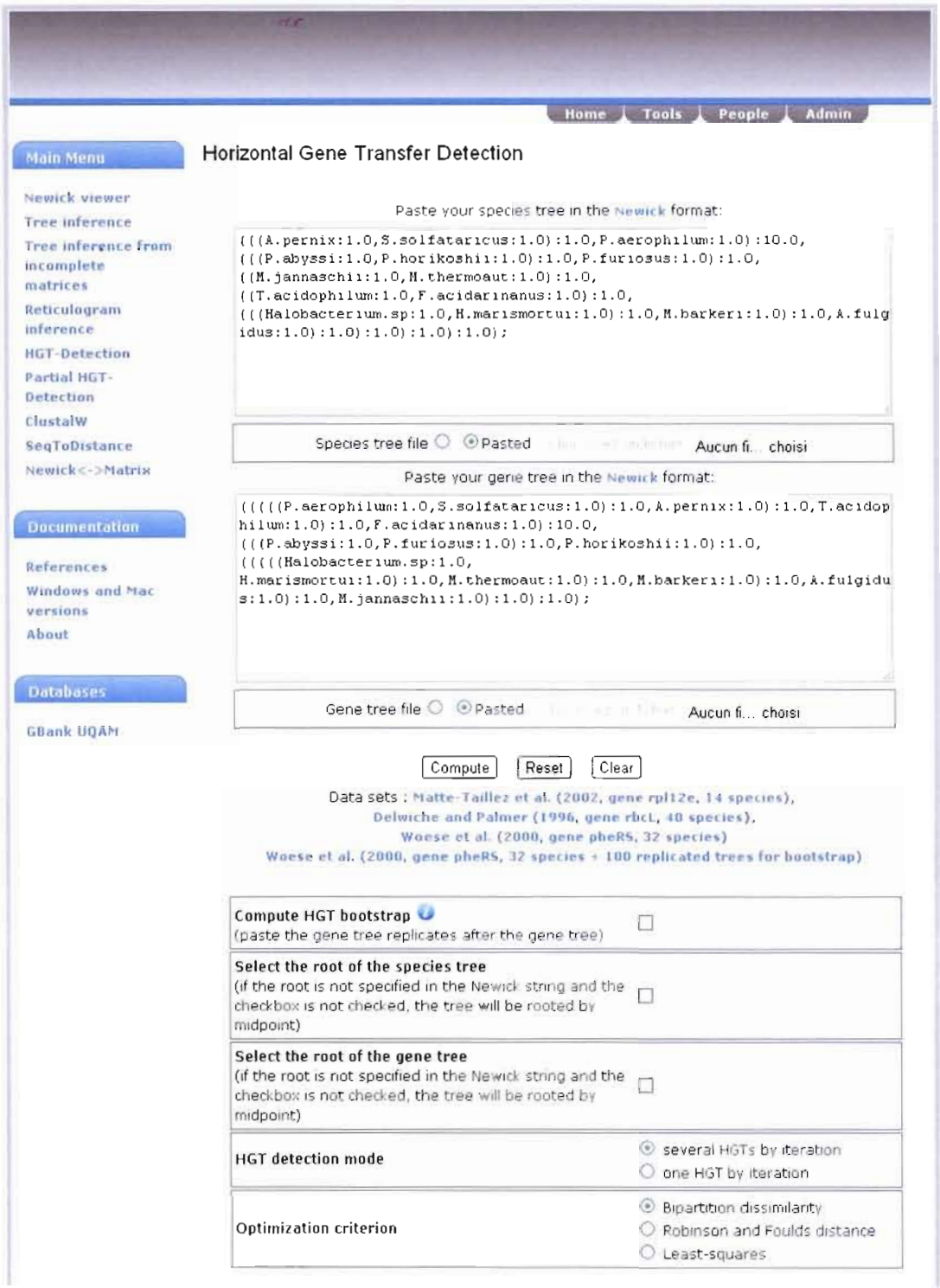


Figure A.2 Interface de *HGT-Detection*.

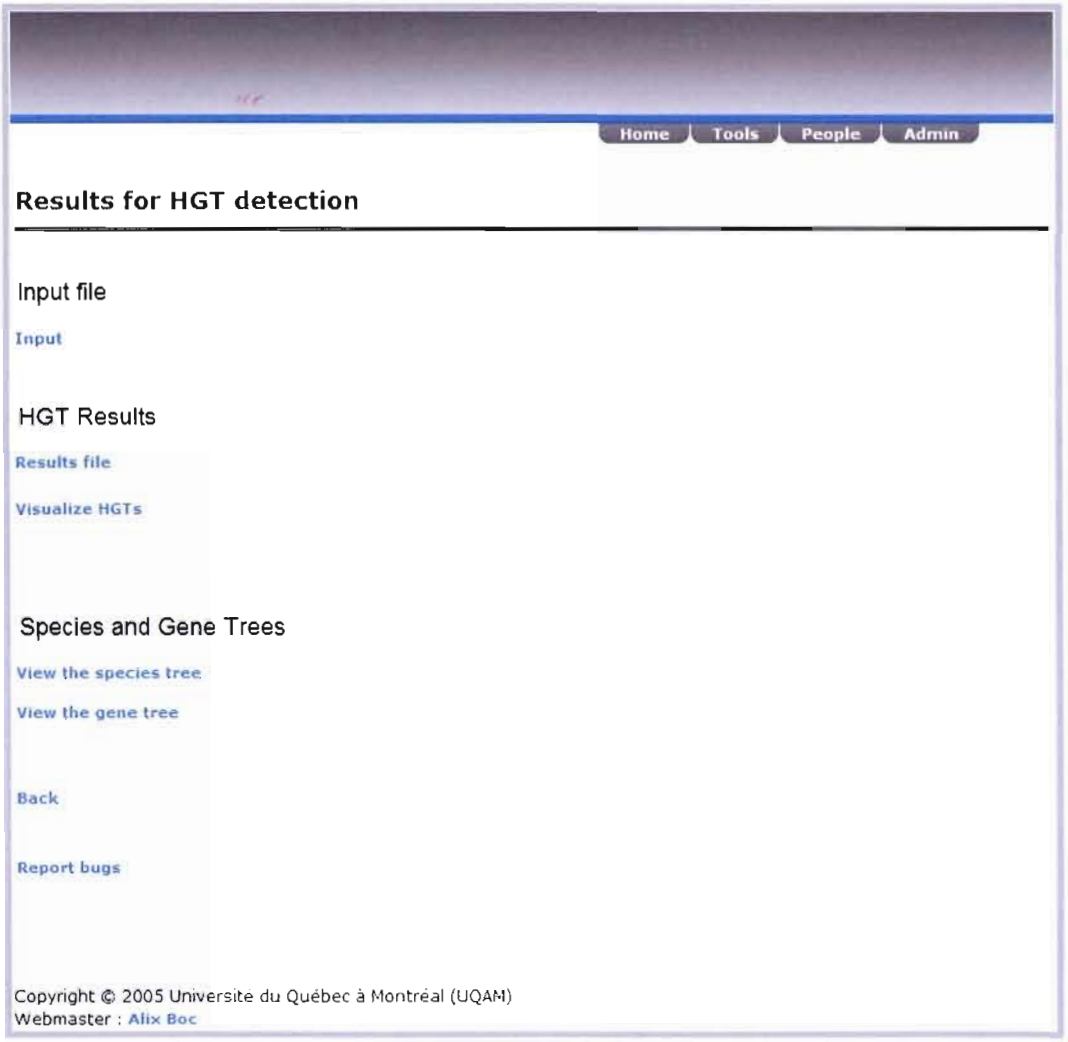


Figure A.3 Page des résultats de *HGT-Detection*.

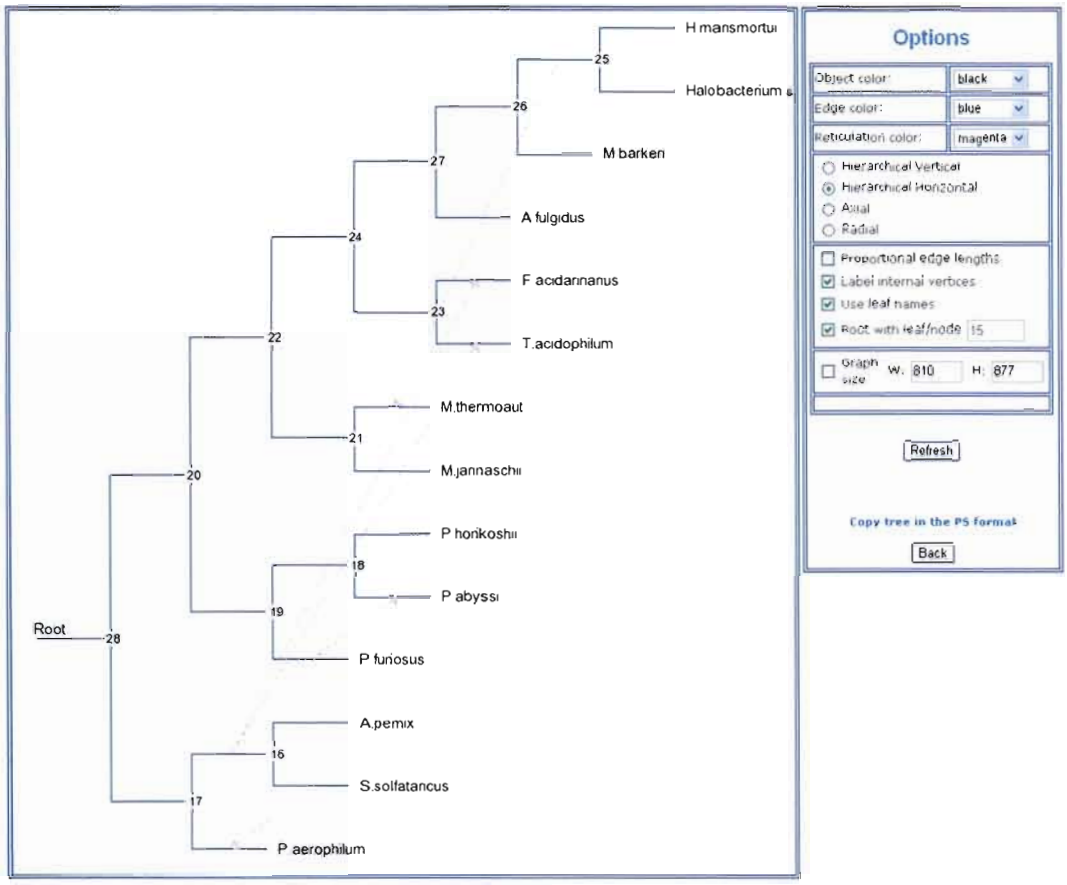


Figure A.4 Exemple de solution à 5 transferts horizontaux, détectés et affichés avec le tracé hiérachique horizontal de l'arbre.

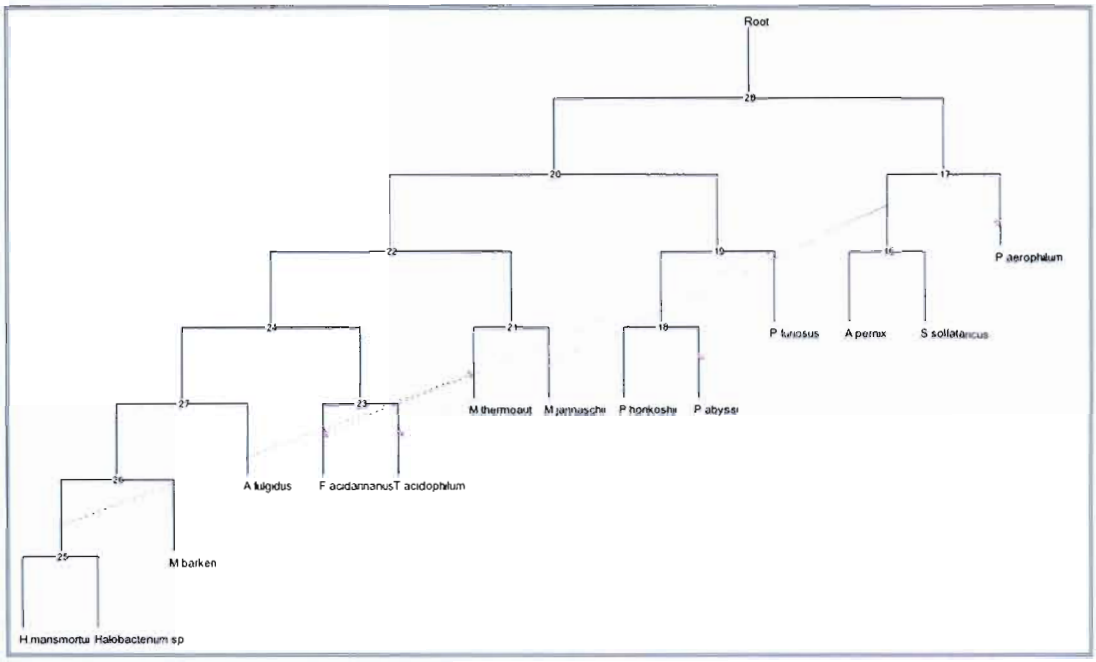


Figure A.5 Exemple de solution à 5 transferts horizontaux, détectés et affichés avec le tracé hiérachique vertical de l'arbre.

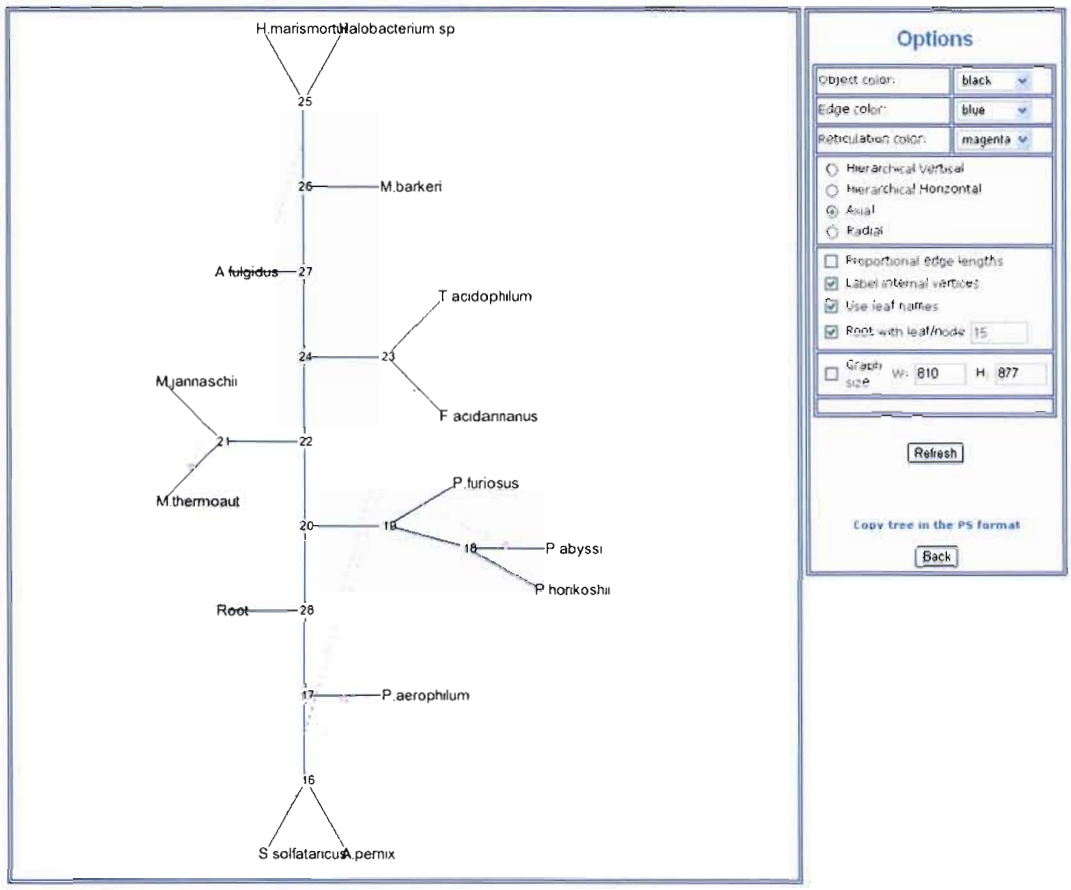


Figure A.6 Exemple de solution à 5 transferts horizontaux, détectés et affichés avec le tracé axial de l'arbre.

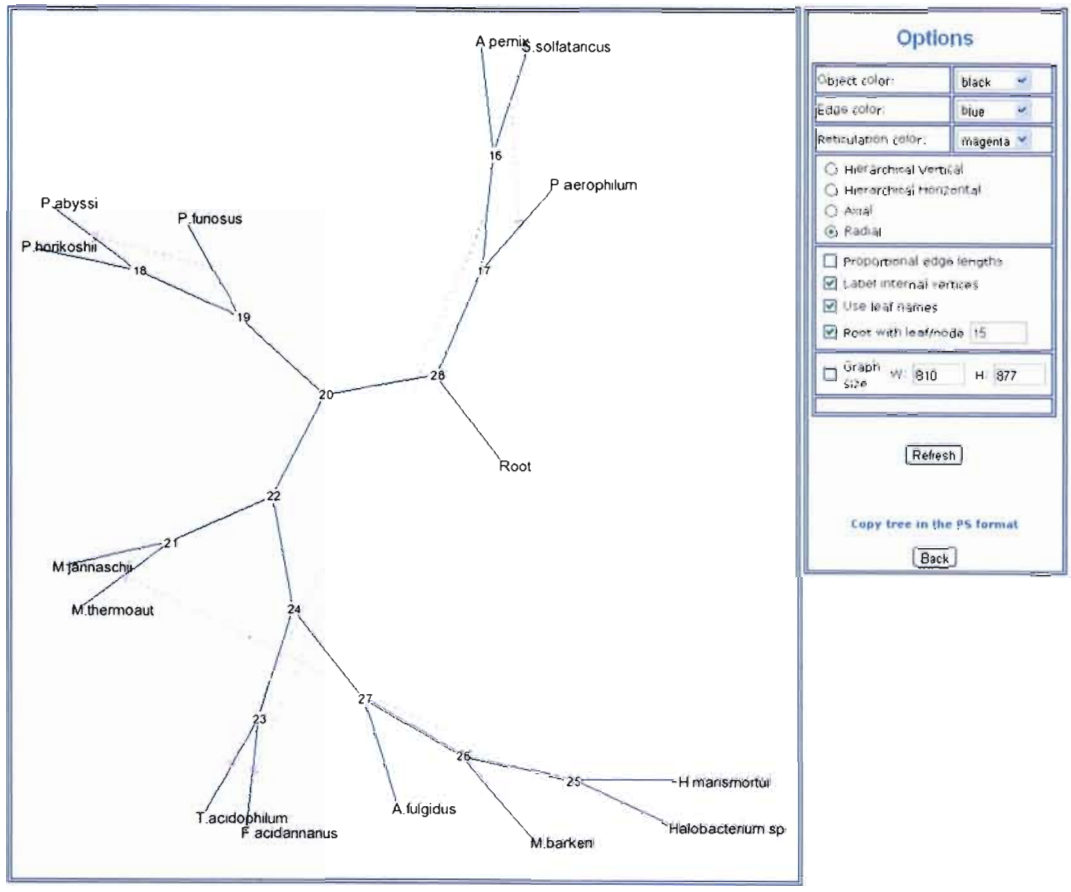


Figure A.7 Exemple de solution à 5 transferts horizontaux, détectés et affichés avec le tracé radial de l'arbre.

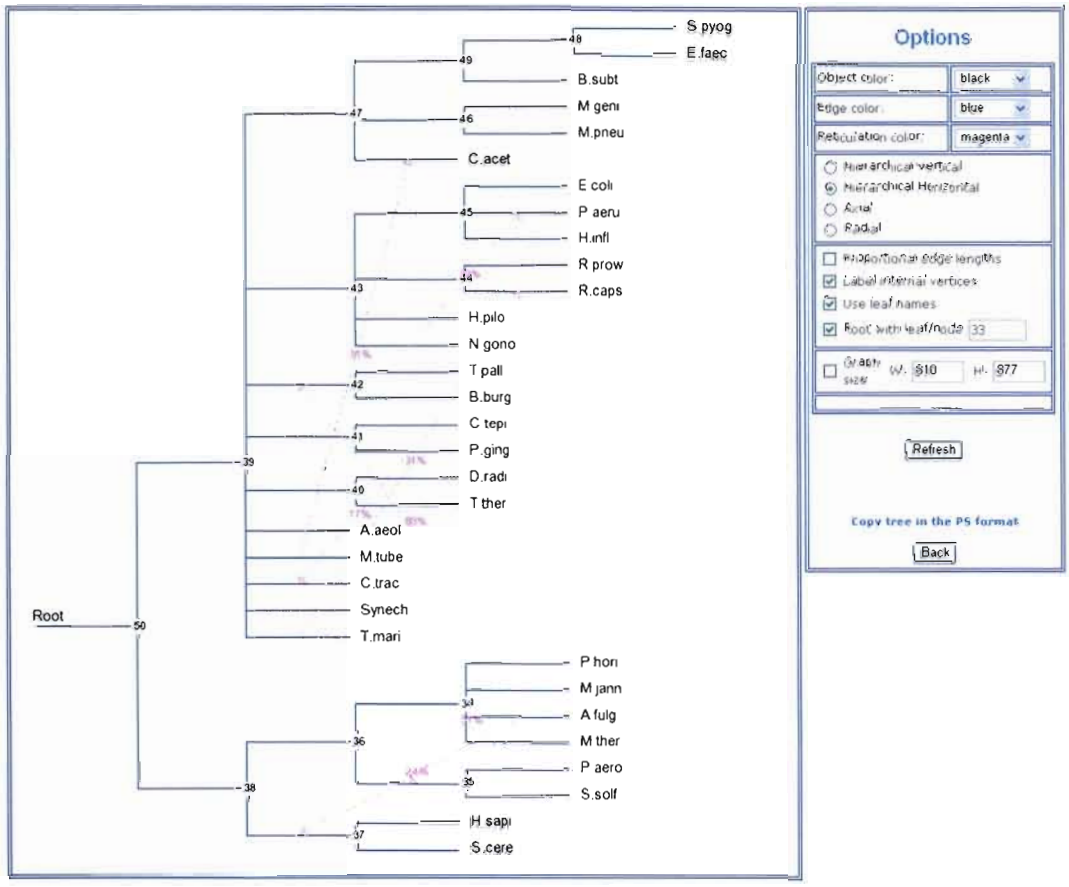


Figure A.8 Exemple de solution à 9 transferts horizontaux, détectés et affichés avec le support de bootstrap en utilisant un tracé hiérachique horizontal de l'arbre.

ANNEXE B : DOCUMENTS SUPPLÉMENTAIRES RELATIFS AU CHAPITRE III

B.1 Schéma algorithmique de *HGT-Detection* (Boc *et al.*, 2010)

Infer species and gene trees T and T' on the same set of species (i.e., leaves);

Root T and T' according to biological evidence or using an outgroup or a midpoint;

if (there exist identical subtrees with two or more leaves in T and T') **then**
Decrease the size of the problem by collapsing them in both T and T';

Select the optimisation criterion OC = LS (least-squares), or RF (Robinson and Foulds distance), or QD (quartet distance), or BD (bipartition dissimilarity);

Compute the initial value of OC between T and T';
$T_0 = T$;
$k = 1$; //k is the Step index

while ($OC \neq 0$)
{
    Find the set of all eligible HGTs (i.e., SPR moves) at step k (denoted by E_HGT_k);
    The set E_HGT_k contains only the transfers satisfying *the subtree constraint*;
      while (*HGTs satisfying the conditions of Theorems 2 and 1 exist*)
      {
          if (there exist HGTs $\in E\_HGT_k$ and satisfying the conditions of Theorems 2) **then**
              Carry out the SPR moves corresponding to these HGTs;

          if (there exist HGTs $\in E\_HGT_k$ and satisfying the conditions of Theorem 1) **then**
              Carry out the SPR moves corresponding to these HGTs;
      }
    Carry out *all remaining SPR moves* corresponding to HGTs satisfying the subtree constraint;
    Compute the value of OC to identify the direction of each HGT;

    $k = k + 1$;
    Decrease the size of the problem by collapsing the identical subtrees in T_k and T';
    Compute the value of OC between T_k and T';
}

Eliminate the idle transfers from the obtained scenario using a backward elimination procedure;

end.

B.2 Trace d'exécution de l'algorithme *RIATA-HGT* appliqué au jeu de données du gène *rpl12e*

```
species tree:
(((A.pernix,S.solfataricus)I10,P.aerophilum)I11,(((P.abyssi,P.horikoshii)I7,P.furiosus)I8,((M.jann
aschii,M.thermoaut.)I5,((T.acidophilum,F.acidarinanus)I3,(((Halobacterium.sp.,H.marismortui)I0,M.b
arkeri)I1,A.fulgidus)I2)I4)I6)I9)I12;

gene tree:
((F.acidarinanus,(((P.aerophilum,S.solfataricus),A.pernix),T.acidophilum)),(((P.horikoshii,P.furio
sus),P.abyssi),(((((Halobacterium.sp.,H.marismortui)I0,M.thermoaut.),M.barkeri),A.fulgidus),M.jann
aschii)));

       There are 3 component(s), which account(s) for 9 solution(s), each of size 5
       --------------------------------------------------------------------------------
       Component I12:
       Subsolution1:
              I0 -> M.thermoaut. (56.0)
              I11 -> F.acidarinanus (100.0)
              I11 -> T.acidophilum (100.0)
       --------------------------------------------------------------------------------
              Component I11:
              Subsolution1:
                     I11 -> A.pernix (74.0) [time violation?]
              Subsolution2:
                     S.solfataricus -> P.aerophilum (74.0)
              Subsolution3:
                     P.aerophilum -> S.solfataricus (74.0)
       --------------------------------------------------------------------------------
              Component I8:
              Subsolution1:
                     P.horikoshii -> P.furiosus (61.0)
              Subsolution2:
                     P.furiosus -> P.horikoshii (61.0)
              Subsolution3:
                     I8 -> P.abyssi (61.0) [time violation?]
       ********************************************************************************
       Consensus network for this set of gene trees
       (((A.pernix,S.solfataricus)I10,P.aerophilum|I11,(((P.abyssi,P.horikoshii|I7,P.furiosus)I8,(
       (M.jannaschii,M.thermoaut.)I5,((T.acidophilum,F.acidarinanus)I3,(((Halobacterium.sp.,H.mari
       mortui)I0,M.barkeri)I1,A.fulgidus)I2)I4|I6)I9)I12;

       P.horikoshii -> P.furiosus
       P.furiosus -> P.horikoshii
       I8 -> P.abyssi
       I11 -> A.pernix
       S.solfataricus -> P.aerophilum
       P.aerophilum -> S.solfataricus
       I0 -> M.thermoaut.
       I11 -> F.acidarinanus
       I11 -> T.acidophilum
```

B.3 Trace d'exécution de l'algorithme *RIATA-HGT* appliqué au jeu de données du gène *PheRS synthétase*

```
species tree:
((((P.hori,M.ther,A.fulg,M.jann)I18,(S.solf,P.aero)I17)I19,(S.cere,H.sapi)I16)I20,((T.ther,D.radi
)I13,(N.gono,H.pilo,(R.caps,R.prow)I10,(P.aeru,(E.coli,H.infl)I11)I14)I12,M.tube,T.mari,(Synech,A.a
eol)I15,(C.trac,(P.ging,C.tepi)I9)I3,(C.acet,((B.subt,(E.faec,S.pyog)I6)I7,(M.pneu,M.geni)I5)I8)I1
)I2,(B.burg,T.pall)I14)I0)I21;
gene tree:
((((D.radi,T.ther)I13,((((N.gono,P.aeru),((R.prow,H.pilo),(H.infl,E.coli)I11)I14,(A.aeol,Synech)I15
)I12.0,((C.trac,P.ging),C.tepi)I9)I3,((R.caps,T.mari),(M.tube,C.acet)I7:4I.0,((M.pneu,M.geni),((S.pyo
g,E.faec),B.subt)I6)I8),((((H.sapi,S.cere)I16,A.fulg),M.ther),(M.jann,((S.solf,P.aero)I17,(P.hori,(T
.pall,B.burg)I14)))))));

       There are 3 component(s), which account(s) for 12 solution(s), each of size 14
       --------------------------------------------------------------------------------
```

```
Component I21:
Subsolution1:
        I17 -> P.hori
        P.hori -> I14 (100.0)
        I16 -> M.ther (25.0)
        I16 -> A.fulg (25.0)
Subsolution2:
        P.hori -> I14 (100.0)
        P.hori -> I17 (85.0)
        I16 -> A.fulg (25.0)
        I16 -> M.ther (25.0)
Subsolution3:
        A.fulg -> I16 (100.0)
        I17 -> M.jann (25.0)
        I17 -> P.hori
        P.hori -> I14 (100.0)
Subsolution4:
        I19 -> M.jann (88.0) [time violation?]
        I16 -> A.fulg (25.0)
        P.hori -> I14 (100.0)
        I16 -> M.ther (25.0)
-------------------------------------------------------------------------------------------
        Component I2:
        Subsolution1:
                R.prow -> H.pilo (67.0)
                R.caps -> T.mari (31.0)
                I4 -> I3 (85.0)
                I11 -> R.prow (0.0)
                I4 -> I13 (100.0)
                R.caps -> M.tube
                P.aeru -> N.gono (55.0)
                M.tube -> C.acet (59.0)
                I4 -> I15 (19.0)
-------------------------------------------------------------------------------------------
                Component I3:
                Subsolution1:
                        P.ging -> C.trac (85.0)
                Subsolution2:
                        C.trac -> P.ging (85.0)
                Subsolution3:
                        I3 -> C.tepi (85.0) [time violation?]
*******************************************************************************
Consensus network for this set of gene trees
((((P.hori,M.ther,A.fulg,M.jann)I18,(S.solf,P.aero)I17)I19,(S.cere,H.sapi)I16)I20,(((T.ther
,D.radi)I13,(N.gono,H.pilo,(R.caps,R.prow)I10,(P.aeru,(E.coli,H.infl)I11)I4)I12,M.tube,T.ma
ri,(Synech,A.aeol)I15,(C.trac,(P.ging,C.tepi)I9)I3,(C.acet,((B.subt,(E.faec,S.pyog)I6)I7,(M
.pneu,M.geni)I5)I8)I1)I2,(B.burg,T.pall)I14)I0)I21;
P.ging -> C.trac
C.trac -> P.ging
I3 -> C.tepi
R.prow -> H.pilo
R.caps -> T.mari
I4 -> I3
I11 -> R.prow
I4 -> I13
R.caps -> M.tube
P.aeru -> N.gono
M.tube -> C.acet
I4 -> I15
I17 -> P.hori
P.hori -> I14
I16 -> M.ther
I16 -> A.fulg
P.hori -> I17
A.fulg -> I16
I17 -> M.jann
I19 -> M.jann
```

[Cette page a été laissée intentionnellement blanche]

ANNEXE C : EXEMPLES DE CODE SOURCE

L'annexe C présente quelques exemples de code source développé pour la détection des transferts complets et partiels. On retrouve ici les deux scripts écrits en langage PERL (*run_hgt.pl* et *run_hgt_partial.pl*) qui encapsulent le programme *HGT-Detection* écrit en langages C/C++. Ces deux scripts permettent de gérer les options du programme, le calcul du bootstrap et le formattage des fichiers de sortie. De plus, le programme principal de *HGT-Detection* (*hgt.cpp*) est présenté, ainsi que la fonction *findBestHGTTab*() qui permet de détecter plusieurs transferts indépendants en une seule itération.

C.1 Script PERL pour la détection des transferts partiels

run_hgt_partial.pl

```
#!/usr/bin/perl

use strict;
use warnings;

print STDOUT "\n\n";
print STDOUT "================================================\n";
print STDOUT "| Partial HGT-DETECTION V.1.0 (October, 2010) |\n";
print STDOUT "| by Alix Boc and Vladimir Makarenkov         |\n";
print STDOUT "================================================\n";

#=====================================================
#= VÉRIFICATION DES ARGUMENTS DE LA LIGNE DE COMMANDE
#=====================================================
if( scalar @ARGV < 1){
    print "Erreur\nusage : $0 speciesTreeFile=[Species tree file]
         geneSequencesFile=[Gene sequences file] opt_m=[PhyML|NJ] opt_ws=[100] opt_ss=[10]
         opt_st=[DNA|RNA|AA] opt_nr=[10] opt_bm=[60]\n";
    print "\nspeciesTreeFile\tSpecies tree file name.
           The species tree should be in the Newick format";
    print "\ngeneSequencesFile\tGene sequences file name.
           The sequences should be in the Phylip format";
    print "\nopt_m \tTree inference (PhyML or NJ)                              : default NJ";
    print "\nopt_ws\tSliding window size                                       : default=100";
    print "\nopt_ss\tStep size                                                 : default=10";
    print "\nopt_st\tSequence type (DNA,AA)                                   : default=AA";
    print "\nopt_nr\tNumber of replicates in bootstrap                        : default=10";
    print "\nopt_bm\tDisplay transfers with bootstrap support higher than x% : default=60%";
    print "\n";
    exit 0;
}

#=============================================================================
#= Gestion des parametres = valeurs par défaut
#=============================================================================
my $fenetre             = 100;    #= taille de la fenêtre
my $increment           = 10;     #= deplacement de la fenêtre
my $nbBootstrap         = 10;     #= nombre d'arbres pour le bootstrap
my $boot_min            = 60;     #= bootstrap acceptable
my $pos_debut_globale   = 0;      #= position de debut
my $reconstruction      = "NJ";   #= NJ,ML
my $sequence_type       = "AA";   #= DNA,AA
my $speciesTreeFile     = "";
my $geneSequencesFile   = "";
```

```
#=============================================================================
#= Gestion des paramètres = nouvelles valeurs
#=============================================================================
my $param;
my @tab_tmp;

for(my $i=0;$i<scalar @ARGV;$i++){
   $param = $ARGV[$i];
   chomp($param);
   @tab_tmp = split("=",$param);
   if($tab_tmp[0] eq "opt_st"){
      $sequence_type = $tab_tmp[1];
      if(($sequence_type ne "DNA") && ($sequence_type ne "AA")){
         print STDOUT "$sequence_type : Unknown sequence type (DNA,AA)\n";
          exit;
       }
    }
   elsif($tab_tmp[0] eq "opt_ws"){
      $fenetre = $tab_tmp[1];
    }
    elsif($tab_tmp[0] eq "opt_m"){
      $reconstruction = $tab_tmp[1];
    }
    elsif($tab_tmp[0] eq "opt_ss"){
      $increment = $tab_tmp[1];
    }
    elsif($tab_tmp[0] eq "opt_bm"){
      $boot_min = $tab_tmp[1];
    }
    elsif($tab_tmp[0] eq "opt_nr"){
      $nbBootstrap = $tab_tmp[1];
    }
    elsif($tab_tmp[0] eq "speciesTreeFile"){
      $speciesTreeFile = $tab_tmp[1];
    }
    elsif($tab_tmp[0] eq "geneSequencesFile"){
      $geneSequencesFile = $tab_tmp[1];
    }
    else{
      print STDOUT $tab_tmp[0] . ": Unknown parameters\n";
      exit;
    }
}

#=============================================================
#= Affichage des données d'exécution
#=============================================================
print STDOUT "Species Tree filename   = $speciesTreeFile\n";
print STDOUT "Gene Sequences filename = $geneSequencesFile\n";
print STDOUT "Sliding windows size    = $fenetre\n";
print STDOUT "Step size               = $increment\n";
print STDOUT "Number of replicates    = $nbBootstrap\n";
print STDOUT "Minimum Bootstrap       = $boot_min\n";
print STDOUT "Tree reconstruction     = $reconstruction\n";
print STDOUT "Sequence Type           = $sequence_type\n";

open (OUT,">phgt_output.txt");
print OUT "=============================================\n";
print OUT "| Partial HGT-DETECTION V.1.0 (October, 2010) |\n";
print OUT "| by Alix Boc and Vladimir Makarenkov         |\n";
print OUT "=============================================\n";
print OUT "Species Tree filename   = $speciesTreeFile\n";
print OUT "Gene Sequences filename = $geneSequencesFile\n";
print OUT "Sliding windows size    = $fenetre\n";
print OUT "Step size               = $increment\n";
print OUT "Number of replicates    = $nbBootstrap\n";
print OUT "Minimum Bootstrap       = $boot_min\n";
print OUT "Tree reconstruction     = $reconstruction\n";
print OUT "Sequence Type           = $sequence_type\n";
close (OUT);

my $phyml     = "exec/phyml";
my $seqboot   = "exec/seqboot";
my $neighbor  = "exec/neighbor";
```

```
my $rf          = "exec/rf";
my $dnadist     = "exec/dnadist";
my $protdist    = "exec/protdist";
my $consense    = "exec/consense";

#=============================================================================================
#= PARTIAL HGT-DETECTION
#=============================================================================================
my $speciesTreeNewick;
my %geneSequences       = ();
my $taille_sequence     = 0;
my $nombre_especes      = 0;
my $tmp;
my $hgtFoundFile        = "hgtdetectes.txt";
my $infile              = "infile";
my $outfile             = "outfile";
my $outtree             = "outtree";
my $intree              = "intree";
my $inputHgtFile        = "input.txt";
my $outputHgtFile       = "output.txt";
my $seqbootConf         = "seqbootConf.txt";    #= fichier de configuration de seqboot
my $dnadistConf         = "dnadistConf.txt";    #= fichier de configuration de dnadist
my $protdistConf        = "protdistConf.txt";   #= fichier de configuration de protdist
my $neighborConf        = "neighborConf.txt";   #= fichier de configuration de neighbor
my $phymlConf           = "phymlConf.txt";      #= fichier de configuration de phyml
my $seed                = 3;                    #= seed pour les progs Phylip
my $log                 = "log.txt";            #= fichier log
my $pos_fin;
my %transferts          = ();
my %transferts2         = ();
my $pos_debut           = $pos_debut_globale;
my $bootTree            = 0;
my $RF_distance         =-1;
my $bootTreeFile        = "bootTreeFile.txt";
my $consenseConf        = "consenseConf.txt";
my $nbNan               = 0;

if( -e "$outputHgtFile.tmp"){
   execute("rm -rf $outputHgtFile.tmp");
}

#=========================================================
#= LECTURE DES DONNÉES (ARBRE ET SÉQUENCES)
#=========================================================
@tab_tmp            = file_get_contents($speciesTreeFile);
$speciesTreeNewick = $tab_tmp[0];

%geneSequences    = chargerSequence($geneSequencesFile);
$nombre_especes   = scalar keys(%geneSequences);
@tab_tmp          = keys(%geneSequences);
$taille_sequence = length($geneSequences{$tab_tmp[0]});

#=========================================================
#= INITIALISATION DES FICHIERS DE CONFIGURATION
#=========================================================
execute("echo \"R\n$nbBootstrap\ny\n$seed\n\" > $seqbootConf");
execute("echo \"M\nd\n$nbBootstrap\ny\n\" > $dnadistConf");
execute("echo \"M\n$nbBootstrap\n$seed\ny\n\" > $neighborConf");
execute("echo \"M\nD\n$nbBootstrap\np\np\np\ny\n\" > $protdistConf");
execute("echo \"y\n\" > $consenseConf");

if($sequence_type eq "DNA"){
   execute("echo \"infile\nb\n$nbBootstrap\ny\ny\n\" > $phymlConf");
   execute("echo \"$infile\n+\n+\nO\nL\n+\nb\n$nbBootstrap\ny\ny\n\" > $phymlConf");
}
else{
   execute("echo \"$infile\nd\n+\nM\nM\nM\n+\nO\nL\n+\nb\n$nbBootstrap\ny\nY\n\" > $phymlConf");
}

print STDOUT "\n=============================================================================================";
print STDOUT "\n|  Interval   |   RF   | Bootstrap | Detected transfers";
print STDOUT "\n=============================================================================================";
my $last          = 0;
my $zone          = 0;
```

```
my $nouvelle_zone = "yes";
my $iteration     = 0;

FOO:
while ( ($pos_debut < $taille_sequence) && ($last == 0)){
   if(($pos_debut + $fenetre) > ($taille_sequence-1)){
      $pos_debut = $taille_sequence - 1 - $fenetre;
      $last = 1;
   }
   printf (STDOUT "\n[ [%3d - %3d]",$pos_debut,$pos_debut + $fenetre);

   $pos_fin = $pos_debut + $fenetre;


   #=====================================================
   #= CRÉATION DU FICHIER INPUT DE SÉQUENCES
   #=====================================================
   open(OUT,">$infile") || die "Cannot open $infile ($!)";
   print OUT "$nombre_especes $fenetre";
   foreach my $elt( keys(%geneSequences)){
      my $espace;
      for(my $i=1;$i<=(10-length($elt));$i++){
         $espace .= " ";
      }
      print OUT "\n" . $elt ."$espace". substr($geneSequences{$elt},$pos_debut,$fenetre);
   }
   close(OUT);

   execute("cat $speciesTreeFile > $inputHgtFile");

   #=====================================================
   #= GÉNÉRATION DES ARBRES
   #=====================================================
   if($reconstruction eq "NJ"){
      execute("rm -rf $outfile >> $log");
      #print STDOUT " seqboot..";
      execute("$seqboot < $seqbootConf >> $log");      #= génération des autres arbres
      execute("cat $outfile > $infile");               #= ajout des autres arbres
      execute("rm -rf $outfile >> $log");
      if($sequence_type eq "DNA"){
         execute("$dnadist < $dnadistConf >> $log");   #= calcul des matrices de distances
         if(!-e $outfile){
            print STDOUT "\nProblems with dnadist...\n";
            exit;
         }
         execute("mv $outfile $infile");
         execute("rm -rf $outtree >> $log");
      }
      else{
         execute("$protdist < $protdistConf >> $log"); #= calcul des matrices de distances
         execute("mv $outfile $infile");
         execute("rm -rf $outtree >> $log");
      }

      $nbNan = `grep "nan" $infile | wc -l`;

      if($nbNan == 0){
         execute("$neighbor < $neighborConf >> $log");   #= transformation en chaine Newick
         execute("cp $outtree $intree");
         execute("echo \"$speciesTreeNewick\" > $inputHgtFile");
         execute("tr -d '\n\r' < $outtree | sed 's/;/;\\\n/g' | sed 's/-//g' >> $inputHgtFile");
         execute("rm -rf $outtree $outfile");
         execute("$consense < $consenseConf >> $log");
         execute("grep 'Sets in' -A 100 $outfile | grep 'Sets NOT' -B 100 | grep '\*' |
                  grep \"[0-9][0-9]*.[0-9][0-9]*\" -o > $bootTreeFile");

         my @tab_tmp=file_get_contents("$bootTreeFile");
         $bootTree=0;
         for(my $i=0;$i<scalar @tab_tmp;$i++){
            $bootTree += ($tab_tmp[$i]*100)/($nbBootstrap+1);
         }
         $bootTree /= scalar @tab_tmp;

         @tab_tmp = file_get_contents2("$inputHgtFile");
```

```
        my $arbre2=$tab_tmp[1];

        $RF_distance = robinson_and_foulds($speciesTreeNewick,$arbre2);
        printf( STDOUT " |  %3d  |    %3.01f%%   | ",$RF_distance,$bootTree);
    }
    else{
        printf( STDOUT " |  Distance matrices cannot be built");
    }
}
else{
    if($reconstruction eq "PhyML"){
        my $cmd = "rm -rf $infile" . "_*";
        execute("$cmd");
        execute("$phyml < $phymlConf >> $log");
        my @tab_tmp = file_get_contents("$infile" . "_phyml_tree.txt");
        my $chaine = $tab_tmp[0];
        (my @tab_boot) = ($chaine =~ /\)([0-9][0-9]*):/g);
        my $total = 0;
        my $boot_cpt=0;
        foreach my $val (@tab_boot){
            $total += $val;
            $boot_cpt++;
        }
        $bootTree = (($total/$boot_cpt)*100)/$nbBootstrap;

        $chaine =~ s/\)[0-9][0-9]*:/\):/g;

        @tab_tmp = file_get_contents("$speciesTreeFile");
        $RF_distance = robinson_and_foulds($tab_tmp[0],$chaine);
        printf( STDOUT " |  %3d  |    %3.01f%%   |",$RF_distance,$bootTree);

        $cmd = "echo \"\n$chaine\" >> $inputHgtFile";
        execute("$cmd");
        @tab_tmp = file_get_contents("$infile" . "_phyml_boot_trees.txt");

        foreach $chaine (@tab_tmp){
            chomp ($chaine);
            $cmd = "echo \"\n$chaine\" >> $inputHgtFile";
            execute("$cmd");
        }
    }
}

#=========================================================
#= DÉTECTION DES TRANSFERTS AVEC BOOTSTRAP
#=========================================================

if(($RF_distance > 1) && ($bootTree >= 0) && ($nbNan == 0 )){
    if($nouvelle_zone eq "yes"){
        $zone++;
    }
    $nouvelle_zone = "no";
    $iteration++;
    execute("rm -rf output.txt outputWeb.txt results.txt nomorehgt.txt log_hgt.txt return.txt");
    execute("perl run_hgt.pl -inputfile=$inputHgtFile -bootstrap=yes >> $log"); #
    execute("cat $inputHgtFile          >> $log");
    execute("cat inputfileformated.txt >> $log");
    execute("cat log_hgt.txt            >> $log");
    execute("cat $outputHgtFile         >> $log");

    #=========================================================
    #= LECTURE DES RÉSULTATS
    #=========================================================
    my @tab_output = file_get_contents($outputHgtFile);
    open(OUT,">>$outputHgtFile.tmp") || die "Cannot open $outputHgtFile.tmp($!)";
    my @tab_list_dest=();
    my @tmp_tab;
    open(IN,"$outputHgtFile") || die "Cannot open $outputHgtFile ($!)";
    foreach my $ligne (@tab_output){
        chomp($ligne);
        $ligne =~ s/ //g;
        print STDOUT "*";
        (my $source,my $dest,my $bootstrap) = split("<>",$ligne);
        @tmp_tab       = split(",",$source);
```

```
        $source        = join(",",@tmp_tab); #join(",",sort par_num @tmp_tab);
        my @tab_source = split(",",$source);
        @tmp_tab       = split(",",$dest);
        $dest          = join(",",@tmp_tab); #join(",",sort par_num @tmp_tab);
        my @tab_dest   =   split(",",$dest);

        if($bootstrap > $boot_min){
           print OUT
                "$iteration<>$RF_distance<>$pos_debut<>$pos_fin<>$source<>$dest<>$bootstrap\n";
           }
        }
        close(IN);
        close(OUT);
     }
  }
  else{
     $nouvelle_zone = "yes";
  }
  if($RF_distance < 2){
     $pos_debut += 2*$increment;
  }

  if(($pos_debut + $fenetre) == ($taille_sequence-1)){
     $pos_debut = $taille_sequence;
  }

  $pos_debut = $pos_debut + $increment;
}

print STDOUT "\n================================================================================\n";


open (OUT,">>phgt_output.txt");
print OUT "\n\nPartial HGT detected with bootstrap support higher than $boot_min% : \n";
print OUT "=============================================================";

my $max_rf = -1;
execute("cat $outputHgtFile.tmp  > $outputHgtFile");
open(IN,"$outputHgtFile") || die "Cannot open $outputHgtFile ($!)";

while(my $ligne = <IN>){
  chomp($ligne);
  if($ligne =~ /<>/){
     $ligne =~ s/ //g;
     (my $iteration,my $RF,$pos_debut,$pos_fin,my $source,my $dest,my $bootstrap) =
     split("<>",$ligne);
     if($max_rf == -1){
        $max_rf = $RF;
     }
     my @tmp_tab = split(",",$source);
     $source  = join(",",@tmp_tab); #join(",",sort par_num @tmp_tab);
     @tmp_tab = split(",",$dest);
     $dest    = join(",",@tmp_tab); #join(",",sort par_num @tmp_tab);
     if($bootstrap > $boot_min){
        my @tmp_tab2 = split("<>",$ligne);
        print OUT "\n\nTransfer  : " . $tmp_tab2[4] ."->". $tmp_tab2[5];
        printf( OUT "\nBootstrap : %1.1lf%%", $tmp_tab2[6]);
        print OUT "\nInterval  : " . $tmp_tab2[2] . "-" . $tmp_tab2[3];

        if(exists($transferts{"$source->$dest"} )){
           for(my $i=$pos_debut;$i<=$pos_fin;$i++){
              $transferts{"$source->$dest"}[$i] = 1;
           }
        }
        else{
           for(my $i=0;$i<$taille_sequence;$i++){
              $transferts{"$source->$dest"}[$i] = 0;
           }
           for(my $i=$pos_debut;$i<=$pos_fin;$i++){
              $transferts{"$source->$dest"}[$i] = 1;
           }
        }
     }
}
```

```
close(IN);

#=============================================
#= AFFICHAGE DES RÉSULTATS
#=============================================

print OUT "\nOverlapping partial HGT detected with bootstrap support higher than $boot_min% :\n";
print OUT "=============================================================================================";
my %results  = ();
my %compteur = ();
my %interval_debut = ();
my %interval_fin = ();
open (IN,$outputHgtFile) || die "Impossible d'ouvrir le fichier $outputHgtFile ($!)";


while( my $ligne=<IN>){
   $ligne =~ s/\[//g;
   $ligne =~ s/\]//g;
   my @tmp = split("<>",$ligne);
   my $key = $tmp[4] . "->" . $tmp[5];

   if(exists($results{$key})){
      if($tmp[6] >= ($boot_min/2)){
         if( ($tmp[2] <= $interval_fin{$key} ) && ($tmp[2] > $interval_debut{$key} ) ){
            $interval_fin{$key} = $tmp[3];
         }
         if( ($tmp[2] > $interval_fin{$key} ) && (($tmp[3]-$interval_fin{$key}) < 40) ){
            $interval_fin{$key} = $tmp[3];
         }
         $results{$key}  = $results{$key} + $tmp[6];
         $compteur{$key} = $compteur{$key} + 1;
      }
   }
   else{
      $results{$key}  = $tmp[6];
      $compteur{$key} = 1;
      $interval_debut{$key}  = $tmp[2];
      $interval_fin{$key}  = $tmp[3];
   }
}

foreach my $key (keys %results){
   if($results{$key}/$compteur{$key} > $boot_min){
      print OUT "\n\nTransfer          : $key";
      printf( OUT "\nAverage bootstrap : %1.1f%%", $results{$key}/$compteur{$key});
      print OUT "\nInterval          : " . $interval_debut{$key} ."-". $interval_fin{$key};
   }
}
print OUT "\n\n";
close (OUT);

open(OUT,">$hgtFoundFile") || die "Impossible d'ouvrir $hgtFoundFile ($!)";
my $saut = "";
my $pred = 0;

foreach my $elt (keys %transferts){
   for(my $i=0;$i<$taille_sequence;$i++){
      if(($transferts{$elt}[$i] == 1) && ($pred == 0)){
         $saut = "\n";
         $pred = 1;
      }
      elsif((($transferts{$elt}[$i] == 0)||($i == ($taille_sequence-1))) && ($pred == 1)){
         $pred=0;
      }
   }
}

close(OUT);

clean();
execute("mv phgt_output.txt output.txt");

print STDOUT "\nSee the results in the file output.txt\n\nEnd of the computation ....\n";
```

```
#=============================================================================================
#================================ SOUS PROGRAMMES ============================================
#=============================================================================================

#-----------------------------------------
#= supprimer un sous-ensemble de feuilles
#-----------------------------------------
sub delete_elt_array{
   (my $elt,my @tab) = @_;
   my @new_tab=();
   foreach my $val (@tab){
      push(@new_tab,$val) if($val ne $elt);
   }
   return @new_tab;
}

#-----------------------------------------
#= exécution du programme externe RF
#-----------------------------------------
sub robinson_and_foulds{
   file_put_contents("rf_input.txt",$_[0] . "\n" . $_[1]);
   execute("$rf rf_input.txt rf_output.txt rf_tmp.txt  rf_output.tmp rf_matrices.txt ");
   my @tab_tmp = file_get_contents("rf_output.txt");
   (my $RF) = ($tab_tmp[5] =~ /.*=\ ([0-9][0-9]*)/);
   return $RF;
}

#-----------------------------------------
#= Chargement du contenu d'un fichier
#-----------------------------------------
sub file_get_contents{
   open(IN,$_[0]) || die "Impossible d'ouvrir " . $_[0] . " ($!)";
   my $i=0;
   my @return = <IN>;
   close(IN);
   return @return;
}

#-------------------------------------------------
#= Chargement du contenu d'un fichier
#= (uniquement les lignes se terminant par ";")
#-------------------------------------------------
sub file_get_contents2{
   open(IN,$_[0]) || die "Impossible d'ouvrir " . $_[0] . " ($!)";
   my $i=0;
   my @return = ();
   while(my $ligne =<IN>){
      if( $ligne =~ ";"){
          $return[$i] = $ligne;
          $i++;
      }
   }
   close(IN);
   return @return;
}

#-----------------------------------------
#= sauvegarde dans un fichier
#-----------------------------------------
sub file_put_contents{
   my $fichier = $_[0];
   open(OUT,">$fichier") || die "Impossible d'ouvrir $fichier ($!)";
   print OUT $_[1];
   close(OUT);
}
#-------------------------------------------------------
#= Encapsulation de l'execution d'une commande externe
#-------------------------------------------------------

sub execute{
   my $cmd = $_[0];
   my $retour = system("$cmd");
}
```

```
#=================================================
#= chargement des séquences dans un tableau associatif
#=================================================
sub chargerSequence{
   my $file = $_[0];
   my $tmp=0;
   my %geneSequences = ();
   open(IN,$file) || die "Cannot open $file ($!)";
   while (my $ligne = <IN>){
      chomp($ligne);
      if($tmp > 0){
         (my $name, my $sequence) = split(" ",$ligne);
         $geneSequences{$name} = $sequence;
      }
      $tmp ++;
   }
   close (IN);
   return %geneSequences;
}
```

C.2 Script PERL pour la détection des transferts complets

```
#!/usr/bin/perl

use strict;
use warnings;

#=========================================================
#= VERIFICATION DES ARGUMENTS DE LA LIGNE DE COMMANDE
#=========================================================
if( scalar @ARGV < 0){
    print "\nErreur\nusage : $0";
    exit 0;
}

my $cmd                  = "exec/hgt ";
my $inputfile            = "";
my $outputfile           = "output.txt";
my $bootstrap            = "no";
my $path                 = "";
my $viewtree="no";
my @tmp_tab;
my @tmp_tab_init;
my %hgt;
my $ligne;
my $nbLines              = 5;
my %hgt_number_tab;
my %hgt_description_tab;
my %hgt_compteur_tab;
my %hgt_criterion_tab;
my %hgt_nbHGT_tab;
my @hgt_pos;
my @hgt_pos2;
my $mode = "";
my $total_hgt;
my $total_trivial;
my $val_retour           = 0;   #= nombre de HGT trouvés
my @hgt_tab;
my $rand_bootstrap       = 0;
my $speciesroot          = "midpoint";
my $generoot             = "midpoint";
my $stepbystep           = "no";

#==== LECTURE DES PARAMETRES ====
foreach my $elt (@ARGV){
   $cmd .= $elt . " ";
   if($elt =~ "bootstrap"){
      @tmp_tab = split("=",$elt);
      $bootstrap = $tmp_tab[1];
      chomp($bootstrap);
```

```
    }
    if($elt =~ "speciesroot"){
       @tmp_tab = split("=",$elt);
       $speciesroot = $tmp_tab[1];
       chomp($speciesroot);
    }
    if($elt =~ "generoot"){
       @tmp_tab = split("=",$elt);
       $generoot = $tmp_tab[1];
       chomp($generoot);
    }
    if($elt =~ "inputfile"){
       @tmp_tab = split("=",$elt);
       $inputfile = $tmp_tab[1];
    }
    if($elt =~ "path"){
       @tmp_tab = split("=",$elt);
       $path = $tmp_tab[1];
    }
    if($elt =~ "viewtree"){
       @tmp_tab = split("=",$elt);
       $viewtree = $tmp_tab[1];
    }
    if($elt =~ "outputfile"){
       @tmp_tab = split("=",$elt);
       $outputfile = $tmp_tab[1];
    }
    if($elt =~ "help"){
       print_description();
       print_help();
       exit;
    }
    if($elt =~ "stepbystep"){
       @tmp_tab = split("=",$elt);
       $stepbystep = $tmp_tab[1];
    }
}

$inputfile             = "$path" . "$inputfile";
$outputfile            = "$path" . "$outputfile";
my $results            = "$path" . "results.txt";
my $tmp_input          = "$path" . "tmp_input.txt";
my $input_no_space     = "$path" . "input_no_space.txt";
my $return_file        = "$path" . "return.txt";
my $log_file           = "$path" . "log_hgt.txt";
my $output_tmp;
my $outputWeb          = "$path" . "outputWeb.txt";
my $generootfile       = "$path" . "geneRootLeaves.txt";
my $speciesrootfile    = "$path" . "speciesRootLeaves.txt";
my $generootfiletmp    = "$path" . "geneRootTmp.txt";
my $speciesrootfiletmp = "$path" . "speciesRootTmp.txt";
my $inputfileformated  = "$path" . "inputfileformated.txt";
my $prehgtfile         = "$path" . "prehgt.txt";

#===== PRINT HEADER =====
print_title();

#= linux like
`rm -rf $results $outputfile $log_file $return_file`;

#===== CHECKING FILES =====
if( $inputfile eq ""){
   print STDOUT "\n\nRUN_HGT : There is no input file";
   exit -1;
}
if( ! -e $inputfile){
   print STDOUT "\n\nRUN_HGT : $inputfile doesn't exist";
   exit -1;
}

if( ($speciesroot eq "file") && ( ! -e $speciesrootfile) ){
   print STDOUT "\n\nRUN_HGT : $speciesrootfile doesn't exist";
   exit -1;
}
```

```
if( ($generoot eq "file") && ( ! -e $generootfile) ){
   print STDOUT "\n\nRUN_HGT : $generootfile doesn't exist";
   exit -1;
}

#=== LECTURE DE L'ARBRE D'ESPECES ===
open(IN,"$inputfile") ||  die "Cannot open $inputfile";
open(OUT,">$inputfileformated") ||  die "Cannot open $inputfileformated";
while($ligne = <IN>){
   chomp($ligne);
   $ligne =~ s/;/;\n/g;
   if($ligne ne ""){
      print OUT $ligne;
   }
}
close(IN);
close(OUT);
open(IN,"$inputfileformated") ||  die "Cannot open $inputfileformated";
my @trees_tab = <IN>;
close(IN);

#=============================================================================
#======================== EXÉCUTION DU PROGRAMME =============================
#=============================================================================
$cmd .= "-inputfile=$tmp_input -outputfile=$outputfile"; # > $log_file";

my $nbTrees = 0 ; # scalar @trees_tab - 1;
#== The program need at least 2 trees
if((scalar @trees_tab < 2)){
   exit_program(-1,$return_file,"PERL : nombre d'arbres invalide");
}
print_minidoc();

FOO:
for (my $i=0;(($i< scalar @trees_tab) && $trees_tab[$i] =~ ";");){
   open(IN,">$tmp_input") ||  die "Cannot open $tmp_input";

   #=============================================================================
   #= In the bootstrap case, we need to change the speciesroot and generoot option
   #= for "file" from the first replicate.
   #=============================================================================
   if($bootstrap eq "yes"){
      if($i == 0){
         print IN $trees_tab[0] . $trees_tab[1];
         $i=2;
         $cmd .= " -randbootstrap=$rand_bootstrap";
      }
      else{
         print IN $trees_tab[0] . $trees_tab[$i++];
      }
      if($i > 2){
         if($cmd !~ /printWeb=no/){
            $cmd .= " -printWeb=no ";
         }
      }
      if($nbTrees == 1){
         $cmd =~ s/-generoot=[a-z][a-z]* //;
         $cmd =~ s/-speciesroot=[a-z][a-z]* //;
         $cmd .= " -generoot=file -speciesroot=file";

         `cp $generootfile $generootfiletmp`;
         `cp $speciesrootfile $speciesrootfiletmp`;
      }
      if($nbTrees > 1){
         `cp $generootfiletmp $generootfile`;
         `cp $speciesrootfiletmp $speciesrootfile`;
      }
   }
   else{
      print IN $trees_tab[$i++] . $trees_tab[$i++];
   }
   close(IN);
```

```
#== traitement des cas de transferts avant la détection
if($stepbystep eq "yes"){
   if(-e $outputWeb){
      my @prehgt;
      my @prehgt_valide;
      my @prehgt_sequence;
      open(PREHGT,"$outputWeb") || die "Probleme avec l'ouverture du fichier $outputWeb";
      my @tab_prehgt = <PREHGT>;
      foreach my $ligne( @tab_prehgt){
         chomp($ligne);
         if($ligne =~ /^hgt_reels =/){
            @prehgt = split(",",$ligne);
         }
         if($ligne =~ /^hgt_valide =/){
            $ligne =~ s/ = /,/;
            @prehgt_valide = split(",",$ligne);
         }
         if($ligne =~ /^hgt_sequence/){
            $ligne =~ s/ = /,/;
            @prehgt_sequence = split(",",$ligne);
         }
      }
      close(PREHGT);
      print STDOUT join("-",@prehgt) . "\n";
      print STDOUT join("-",@prehgt_valide) . "\n";
      print STDOUT join("-",@prehgt_sequence) . "\n";

      open(PREHGT,">$prehgtfile") || die "Impossible d'ouvrir $prehgtfile";
      print STDOUT "\nnbHGT  = " . $#prehgt_valide . "\n\n";

      for(my $i=1;$i< scalar(@prehgt_valide);$i++){
         print PREHGT $prehgt_sequence[$i] . " " . $prehgt[4*$i-3] . " " .
                      $prehgt[4*$i-2] . " " . $prehgt[4*$i-1] . " " .
                      $prehgt[4*$i-0] . " " . $prehgt_valide[$i] . "\n";
      }
      close(PREHGT);
   }
}

print STDOUT "\nComputation " . ++$nbTrees . " in progress...";
execute_hgt("$cmd >> $log_file");
if( ! -e $results){
   print STDOUT "\n\nRUN_HGT : An error has occured during computation.
                 Check the log file ($log_file) for more details ! ";
   exit -1;
}
print STDOUT "formatting results...";

if((($i == 2) && ($viewtree eq "yes")) || (($i == 1) && ($viewtree eq "yes"))){
   exit_program(0,$return_file,"RUN_HGT : We just want to see the input trees");
}

#=============================================================================
#========================== LECTURE DES RÉSULTATS ============================
#=============================================================================
open(IN,"$results") ||  die "\nCannot open $results ! !!";
@hgt_tab = <IN>;
close(IN);

exit_program(-1,$return_file,"RUN_HGT : result file empty") if(scalar @hgt_tab == 0);

$mode = $hgt_tab[1] if(exists($hgt_tab[1]));
chomp($mode);

if(($i == 2) || ($bootstrap eq "no")){
   print STDOUT "done";
   my $nbHGT=0;
   my $nbHGT2=0;
   for(my $j=2;$j<= scalar @hgt_tab;){
      if($hgt_tab[$j] =~ /^[0-9]/){
         my $cpt = read_line($hgt_tab[$j++]);
         for(my $k=0;$k<$cpt;$k++){
            my $hgt_number = read_line($hgt_tab[$j++]);
            my $source_list = read_line($hgt_tab[$j++]);
```

```
            my $dest_list = read_line($hgt_tab[$j++]);
            my $transfer_description2 = read_line($hgt_tab[$j++]);
            my $transfer_description = "From subtree ($source_list) to subtree ($dest_list)";

            my $criterion_list = read_line($hgt_tab[$j++]);
            my $criterion_list2 = read_line($hgt_tab[$j++]);

            $hgt_number_tab{"$source_list<>$dest_list"}      = $hgt_number;
            $hgt_description_tab{"$source_list<>$dest_list"} = $transfer_description;
            $hgt_compteur_tab{"$source_list<>$dest_list"}    = 1;
            $hgt_criterion_tab{"$source_list<>$dest_list"}   = $criterion_list;
            $hgt_nbHGT_tab{"$source_list<>$dest_list"}       = $cpt;
            $hgt_pos[$nbHGT++] = "$source_list<>$dest_list";
         }
         if($mode eq 'mode=multicheck'){
            $hgt_pos2[$nbHGT2++] = read_line($hgt_tab[$j++]);
         }
      }
      else{
         @tmp_tab_init = split(",",$hgt_tab[0]);
         @tmp_tab = split(" ",$hgt_tab[$j]);
         $total_hgt = $tmp_tab[1];
         $total_trivial = $tmp_tab[2];
         if($bootstrap eq "no"){
            open(OUT,">>$outputfile") || die "Cannot open $outputfile";
            if($total_hgt > 0){
               print_result();
            }
            else{
               print OUT " : no HGTs have been found !";
            }
            close(OUT);
            exit_program($val_retour,$return_file,"PERL : pas de bootstrap,
                        on traite un seul input");
            %hgt_number_tab=();
            %hgt_description_tab=();
            %hgt_compteur_tab=();
            %hgt_criterion_tab=();
            %hgt_nbHGT_tab=();
            @hgt_pos=();
            @hgt_pos2=();
         }
         $j = (scalar @hgt_tab) + 1;
      }
   }
   if(($i == 2) && ($bootstrap eq "yes") && $nbHGT == 0){
      last FOO;
   }
}
if(($bootstrap eq "yes") && ($i > 2)){
   print STDOUT "done";
   my $nbHGT=0;
   my $nbHGT2=0;
   for(my $j=2;$j<= scalar @hgt_tab;){
      if($hgt_tab[$j] =~ /^[0-9]/){
         my $cpt = read_line($hgt_tab[$j++]);
         for(my $k=0;$k<$cpt;$k++){
            my $hgt_number = read_line($hgt_tab[$j++]);
            my $source_list = read_line($hgt_tab[$j++]);
            my $dest_list = read_line($hgt_tab[$j++]);
            my $transfer_description2 = read_line($hgt_tab[$j++]);
            my $transfer_description = "From subtree ($source_list) to subtree ($dest_list)";
            my $criterion_list = read_line($hgt_tab[$j++]);
            if(exists $hgt_compteur_tab{"$source_list<>$dest_list"}){
               $hgt_compteur_tab{"$source_list<>$dest_list"}   += 1;
            }
         }
         if($mode eq 'mode=multicheck'){
            $j++;
         }
      }
      else{
         $j = (scalar @hgt_tab) + 1;
      }
```

```
      }
   }
}

if($bootstrap eq "yes"){
   open(OUT,">>$outputfile") || die "Cannot open $outputfile";
   print_result2();
   close(OUT);
}

exit_program($val_retour,$return_file,"PERL : fin normale du programme");


#================================================================================
#============================== FONCTIONS =======================================
#================================================================================

sub read_line{
  my ($line) = @_;
  chomp($line);
  return $line;
}

sub exit_program{
  my($val,$file,$message) = @_;
  open(RET,">$file") || die "Cannot open $file";
  print RET $val;
  close(RET);
  print STDOUT "\n";
  #print STDOUT "\nexit=>$message";
  exit;
}

sub execute_hgt{
    my ($cmd) = @_;
    my $retour = 0;
   #print STDERR "\n$cmd";
    system($cmd);
}

sub print_result2{
   my $first=0;
   foreach my $elt(@hgt_pos){
      print OUT "\n" if($first > 0);
      print OUT "$elt";
      printf(OUT "<>%3.11f",$hgt_compteur_tab{$elt}*100/$nbTrees,$hgt_compteur_tab{$elt},
$nbTrees); #= if($hgt_number_tab{"$elt"} !~ "Trivial");
      $first=1;
   }
}

sub print_result{
    my $nbHGT2=0;
    my $newGroup=0;
    my $cpt=1;
    my @tmp_tab = split(",",$hgt_tab[0]);

   print OUT   "================================================================================\n";
   print OUT   "| Program : HGT Detection 3.2 - November, 2009                                  |\n";
   print OUT   "| Authors   : Alix Boc and Vladimir Makarenkov (UQAM)                           |\n";
   print OUT   "| This program computes a unique scenario of horizontal gene transfers (HGT) |\n";
   print OUT   "| for the given pair of species and gene phylogenetic trees.                   |\n";
   print OUT   "================================================================================\n";

    print OUT "\nSpecies tree :\n". $trees_tab[0] . "\nGene Tree :\n" . $trees_tab[1];

    print OUT "\n\n=============================================";
     print OUT "\n= Criteria values before the computation ";
     print OUT "\n=============================================";
   if($bootstrap eq "yes"){
     printf (OUT "\nRobinson and Foulds distance (RF) = %d",$tmp_tab_init[0]);
     printf (OUT "\nLeast-squares coefficient(LS)     = %1.3lf",$tmp_tab_init[1]);
     printf (OUT "\nBipartition dissimilarity         = %1.11f\n",$tmp_tab_init[2]);
   }
```

```
else{
  printf (OUT "\nRobinson and Foulds distance (RF) = %d",$tmp_tab[0]);
  printf (OUT "\nLeast-squares coefficient(LS)     = %1.3lf",$tmp_tab[1]);
  printf (OUT "\nBipartition dissimilarity         = %1.11f\n",$tmp_tab[2]);
}

 if($bootstrap eq "yes"){
    printf(OUT "\n\nBootstrap values were computed with %d gene trees",$nbTrees);
 }
 print OUT "\n\n";
 foreach my $elt( @hgt_pos){
     if(($newGroup == 0 ) && ($mode eq 'mode=multicheck')){
         print OUT "\n=============================================================================";
         if($hgt_nbHGT_tab{"$elt"} == 1){
             print OUT "\n| Iteration #$cpt : ". $hgt_nbHGT_tab{"$elt"} ." HGT was found";
         }
         else{
           print OUT "\n| Iteration #$cpt : ". $hgt_nbHGT_tab{"$elt"} ." HGTs were found";
         }

         print OUT "\n=============================================================================";
         print OUT "\n|";
         $newGroup = 1;
         $cpt++;
     }
     else{
         if($mode eq 'mode=monocheck'){
             print OUT "\n=============================================================================";
         }
     }
     print OUT "\n| "  . $hgt_number_tab{"$elt"};

     if(($bootstrap eq "yes")&&($hgt_number_tab{"$elt"} !~ "Trivial")){
         printf(OUT "(bootstrap value = %3.11f%%) ",$hgt_compteur_tab{$elt}*100/$nbTrees,
                                                    $hgt_compteur_tab{$elt}, $nbTrees);
     }
     print OUT "\n| "  . $hgt_description_tab{"$elt"};
     print OUT "\n| "  . $hgt_criterion_tab{"$elt"};
     if($mode eq 'mode=monocheck'){
         print OUT "\n=============================================================================\n";
     }
     else{
         print OUT "\n| ";
     }
     my $tmp = "HGT " . $hgt_nbHGT_tab{"$elt"} . " / " . $hgt_nbHGT_tab{"$elt"} . " ";
     my $tmp2 = $tmp . " Trivial";

     if(($mode eq 'mode=multicheck') &&
       (( $tmp =~ $hgt_number_tab{"$elt"}) ||($tmp2 =~ $hgt_number_tab{"$elt"}))){
         print OUT "\n=============================================================================";
         print OUT "\n| After this iteration the criteria values are as follows :";
         print OUT "\n| " . $hgt_pos2[$nbHGT2++] ;
         print OUT "\n=============================================================================\n";
         $newGroup=0;
     }
 }

 print OUT "\nTotal number of HGTs : $total_hgt ";
 if( $total_trivial > 0){
    print OUT "(". ($total_hgt-$total_trivial) ." regular + " .
                                                  $total_trivial . " trivial HGTs)";
 }
 $val_retour = $total_hgt;

open(OUTWEB,">>$outputWeb");
if($bootstrap eq "yes"){
   print OUTWEB "\nbootHGT=";
   my $first=0;
   foreach my $elt( @hgt_pos){

      if(($bootstrap eq "yes")&&($hgt_number_tab{"$elt"} !~ "Trivial")){
         if($first==0){
            printf(OUTWEB "%3.01f",$hgt_compteur_tab{$elt}*100/$nbTrees);
         }
```

```
            else{
               printf(OUTWEB ",%3.01f",$hgt_compteur_tab{$elt}*100/$nbTrees);
            }
            $first=1;
         }
      }
   }
   close OUTWEB;
}


sub print_title{
   print STDOUT "=================================================================================\n";
   print STDOUT "| HGT-DETECTION V.3.2 (November, 2009) by Alix Boc and Vladimir Makarenkov |\n";
   print STDOUT "=================================================================================\n";
}

sub print_minidoc{
   print STDOUT "\nCheck the file $log_file for the computation details";
   print STDOUT "\nCheck the file $outputfile for the program output\n";
}

sub print_description{
   print STDOUT  "=================================================================================\n";
   print STDOUT  "| Program : HGT Detection 3.2 - November, 2009                                  |\n";
   print STDOUT  "| Authors   : Alix Boc and Vladimir Makarenkov (UQAM)                           |\n";
   print STDOUT  "| This program computes a unique scenario of horizontal gene transfers for|\n";
   print STDOUT  "| the given pair of species and gene phylogenetic trees.                       |\n";
   print STDOUT  "=================================================================================\n";
}

sub print_help{
   print STDOUT "\nUsage :\nperl run_hgt.pl -inputfile=[inputfilename] -
                 outputfile=[outputfilename] -criterion=[rf|ls|bd]";
   print STDOUT "-speciesroot=[midpoint|file] -generoot=[midpoint|file|bestbipartition]";
   print STDOUT "-scenario=[unique|multiple] -nbhgt=[maxhgt] -path=[path] -bootstrap=[no|yes]";
   print STDOUT "\n\nsee README.txt file for more detail.";
}
```

C.3 Programme principal pour la détection des transferts complets

```
//=============================================================================================
//=  HGT-DETECTION v3.2b
//=  Authors : Alix Boc and Vladimir Makarenkov
//=  Date    : November 2009
//=
//=  Description : This program detect horizontal gene transfers (HGTs). As input it takes two
//=  trees : a species tree and a gene tree. The goal is to transform the species tree
//=  into the gene tree by series of SPR operations. There are 3 criteria : the Robinson and
//=  Foulds distance, the least-square criterion and the bipartition dissimilarity.
//=  We also use the subtree constraint.
//=
//=  input   : file with species tree and gene tree in the Newick format.
//=            In the case of simulations, the species tree and all the gene trees should be
//=            in the same file in the Phylip format or submitted as a Newick string
//=  output  : a list of HGT and the criteria values for each transfer.
//=
//=============================================================================================

#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <string.h>
#include <time.h>
#include <signal.h>

#pragma warning(disable:4996)

#include "structures.h"
#include "utils_tree.cpp"
#include "fonctions.cpp"

#define binaireSpecies 0
#define binaireGene    1

void traiterSignal(int sig){
   printf("\nMESSAGE : SEGMENTATION FAULT #%d DETECTED",sig);
   printf("\nUse valgrind or gdb to fix the problem");
   printf("\n");
   exit(-1);
}

//=============================================================================================
//=============================================== MAIN ========================================
//=============================================================================================
int main(int nargc,char **argv){

   struct InputTree SpeciesTree;              //== initial species tree
   struct InputTree SpeciesTreeCurrent;       //== initial species tree
   struct InputTree FirstTree;
   struct InputTree FirstTree2;
   struct InputTree GeneTree;                 //== initial gene tree
   struct InputTree SpeciesTreeRed;           //== reduced species tree
   struct InputTree GeneTreeRed;              //== reduced gene tree
   struct ReduceTrace aMap;                   //== mapping structure between species tree and
                                                   reduced species tree
   struct InputTree geneTreeSave;
   struct HGT * bestHGTRed = NULL;            //== list of HGTs for the reduced tree
   struct HGT * bestHGT = NULL;               //== list of HGTs for the regular tree
   struct HGT * outHGT = NULL;
   struct HGT * bestHGTmulticheck = NULL;
   int nbHGT_boot;
   int first = 1,k,l;
   int cpt_hgt,i,j,tmp,nbTree=0;
   int bootstrap = 0;
   int multigene = 0;
   int nbHgtFound = 0;
   struct CRITERIA * multicheckTab=NULL;
   struct CRITERIA aCrit;                    //== struture of all the criteria
   struct DescTree *DTSpecies,               //== structure of submatrices for the species tree
                   *DTGene;                  //== structure of submatrices for the gene tree
   struct Parameters param;
```

```
   FILE *in,*out;
   int max_hgt,nbHGT;
   int ktSpecies;
   int trivial = 1;
   int *speciesLeaves = NULL;
   int RFref;
   int imc;
   char *mot = (char*)malloc(100);
   int nomorehgt=0;
   initInputTree(&geneTreeSave);


   //== read parameters
   printf("\nhgt : reading options");
   if(readParameters(&param,argv,nargc)==-1){
      printf("\nhgt : no options specified, see the README file for more details\n");
      exit(-1);
   }

   rand_bootstrap = param.rand_bootstrap;

   signal(SIGSEGV,traiterSignal);

   //== open the input file
   if((in=fopen(param.inputfile,"r"))==NULL){
      printf("\nhgt : The file %s does not exist",param.inputfile);
      exit(-1);
   }
   if(strcmp(param.speciesroot,"file") == 0){
      if(!file_exists(param.speciesRootfileLeaves)){
         printf("\nhgt : The file %s does not exist",param.speciesRootfileLeaves);
         exit(-1);
      }
   }
   if(strcmp(param.generoot,"file") == 0){
      if(!file_exists(param.geneRootfileLeaves)){
         printf("\nhgt : The file %s does not exist",param.geneRootfileLeaves);
         exit(-1);
      }
   }
   if((in=fopen(param.inputfile,"r"))==NULL){
      printf("\nhgt : Cannot open input file (%s)",param.inputfile);
      exit(-1);
   }

   //== open the bootstrapFile
   if(strcmp(param.bootstrap,"yes") == 0){
      bootstrap = 1;
   }
   if(strcmp(param.multigene,"yes") == 0){
      multigene = 1;
      if(strcmp(param.speciesroot,"file"))
         strcpy(param.speciesroot,"midpoint");
   }
   remove(param.hgtResultFile);

   initInputTree(&FirstTree);
   initInputTree(&SpeciesTreeCurrent);

   FILE * results, *results2;
   if((results = fopen(param.results,"w+"))==NULL){
      printf("\nhgt : Cannot open input file (%s)",param.results);
      exit(0);
   }
   if((results2 = fopen(param.results2,"w+"))==NULL){
      printf("\nhgt : Cannot open input file (%s)",param.results2);
      exit(0);
   }


//=============================================================================
//============================ LECTURE DES ARBRES =============================
//=============================================================================
```

```
   printf("\nhgt : reading the input file");
   tmp = readInputFile(in, param.input/*,&SpeciesTree,&GeneTree*/,param.errorFile);

   if(tmp==-1) {
      printf("\nCannot read input data !!\n");
      exit(-1);
   }


   cpt_hgt = 0;

   initInputTree(&SpeciesTree);
   initInputTree(&GeneTree);
   initInputTree(&SpeciesTreeRed);
   initInputTree(&GeneTreeRed);

   //== lecture des matrices ou des chaines Newick en entréee
   if(readInput(SPECIE,param.input,&SpeciesTree) == -1){
      printf("\nError in species tree\n"); exit(-1);
   }
   if(readInput(GENE,param.input,&GeneTree) == -1){
      printf("\nError in gene tree\n"); getchar(); exit(-1);
   }

   TrierMatrices(GeneTree.Input,GeneTree.SpeciesName,SpeciesTree.SpeciesName,SpeciesTree.size);

   NJ(SpeciesTree.Input,SpeciesTree.ADD,SpeciesTree.size);
   NJ(GeneTree.Input,GeneTree.ADD,GeneTree.size);

   //== Construction des differentes représentations des arbres (adjacence,arêtes,longueurs,degré)
   CreateSubStructures(&SpeciesTree,1,binaireSpecies);
   CreateSubStructures(&GeneTree,1,binaireGene);


//=============================================================================
//============================= GESTION DES RACINES ===========================
//=============================================================================
   //== sélection de la racine
   printf("\nhgt : adding the tree roots");
   if(strcmp(param.load,"yes") == 0 ){
      chargerFichier(&SpeciesTree,param.speciesTree,param.speciesRootfile);
      chargerFichier(&GeneTree,param.geneTree,param.geneRootfile);
   }
   else{
      if((strcmp(param.version,"web")==0) && (strcmp(param.printWeb,"yes")==0)){
          saveTree(param.speciesTreeWeb,SpeciesTree,bestHGT,0,cpt_hgt,"",param.scenario,NULL);
          saveTree(param.geneTreeWeb,GeneTree,bestHGT,0,cpt_hgt,"",param.scenario,NULL);
      }
      if(first == 1){
         int nbBranche,leave;
      }
      if(speciesLeaves == NULL){
         speciesLeaves = (int*) malloc(SpeciesTree.size*sizeof(int));
         speciesLeaves[0] = -1;
      }

      if(SpeciesTree.Root == -1)
         addRoot(&SpeciesTree,NULL,SpeciesBranch,param.speciesroot,
                 param.speciesRootfile,param.speciesRootfileLeaves,NULL,param.version);
         if(GeneTree.Root == -1)
            addRoot(&GeneTree,NULL,GeneBranch,param.generoot,param.geneRootfile,
                    param.geneRootfileLeaves,NULL,param.version);

         if(strcmp(param.viewtree,"yes")==0)
            exit(0);
      }

      nbTree++;

      if(SpeciesTree.size > GeneTree.size) max_hgt = 4*GeneTree.size * GeneTree.size;
      else max_hgt = 4*SpeciesTree.size * SpeciesTree.size;

      bestHGTRed = (struct HGT*)malloc(max_hgt*sizeof(struct HGT));
      bestHGT = (struct HGT*)malloc(max_hgt*sizeof(struct HGT));
```

```
for(i=0;i<max_hgt;i++){
   bestHGTRed[i].listSource = NULL;
   bestHGTRed[i].listDestination = NULL;
   bestHGT[i].listSource = NULL;
   bestHGT[i].listDestination = NULL;
}

if(first==1 && strcmp(param.scenario,"multiple")!=0 && strcmp(param.mode,"multicheck")==0)
multicheckTab = (struct CRITERIA *)malloc(max_hgt*sizeof(struct CRITERIA));

if(first==1){
   multicheckTab[0].m = 0;
   imc=0;
}
InitCriteria(&aCrit,SpeciesTree.size);
DTSpecies = (struct DescTree*)malloc((2*SpeciesTree.size-2-SpeciesTree.kt+1)*
                                     sizeof(struct DescTree));

RechercherBipartition(SpeciesTree.ARETE,SpeciesTree.ADD,SpeciesTree.Root,
                      SpeciesTree.Adjacence,DTSpecies,SpeciesTree.size,SpeciesTree.kt);

if(first ==1){
   FirstTree.ADD=NULL;
   FirstTree.ARETE=NULL;
   copyInputTree(&FirstTree,SpeciesTree,1,1);
   AdjustBranchLength(&FirstTree,GeneTree,binaireSpecies,1);
}

InitCriteria(&aCrit,SpeciesTree.size);
computeCriteria(FirstTree.ADD,GeneTree.ADD,FirstTree.size,&aCrit,
                FirstTree.LONGUEUR,FirstTree.ARETE,GeneTree.LONGUEUR,GeneTree.ARETE);

AdjustBranchLength(&SpeciesTree,GeneTree,binaireSpecies,1);

if((bootstrap != 1) || (bootstrap==1 && first==1)){
   fprintf(results,"%d,%lf,%lf\n",aCrit.RF,aCrit.LS,aCrit.BD);
   RFref=aCrit.RF;
}

//=========================================================================
//== Ajout de transferts avant le processus de détection
//=========================================================================
//printf("\nhgt : stepbystep=%s",param.stepbystep);
//if(strcmp(param.stepbystep,"yes") == 0){
//== si le fichier prehgtfile existe,
if(file_exists(param.prehgtfile)){
   FILE *prehgt = fopen(param.prehgtfile,"r");
   printf("\nhgt : adding pre-hgt");
   int pos_source,pos_dest,step,valide,newStep=-1;
   int a1,a2,b1,b2;
   int dedans=0,dedans2=0;
   int nbHGTadd=0;
   while(fscanf(prehgt,"%d%d%d%d%d%d",&step,&a1,&a2,&b1,&b2,&valide) != -1){
      dedans=0;
      SpeciesTreeCurrent.ADD=NULL;
      SpeciesTreeCurrent.ARETE=NULL;
      copyInputTree(&SpeciesTreeCurrent,SpeciesTree,1,1);

      DTSpecies = (struct DescTree*)malloc((2*SpeciesTree.size-2-SpeciesTree.kt+1)*
                                           sizeof(struct DescTree));
      RechercherBipartition(SpeciesTree.ARETE,SpeciesTree.ADD,SpeciesTree.Root,
                            SpeciesTree.Adjacence,DTSpecies,SpeciesTree.size,SpeciesTree.kt);

      if (valide == 2){
         int tmp=a1; a1=b1; b1=tmp;
         tmp=a2; a2=b2; b2=tmp;
         valide=1;
      }
      pos_source = pos_dest = -1;
      for(i=1;i<=2*SpeciesTree.size-3-SpeciesTree.kt;i++){
         if((SpeciesTree.ARETE[2*i-1] == a1 && SpeciesTree.ARETE[2*i-2] == a2) ||
            (SpeciesTree.ARETE[2*i-2] == a1 && SpeciesTree.ARETE[2*i-1] == a2))
            pos_source = i;
```

```
            if((SpeciesTree.ARETE[2*i-1] == b1 && SpeciesTree.ARETE[2*i-2] == b2) ||
               (SpeciesTree.ARETE[2*i-2] == b1 && SpeciesTree.ARETE[2*i-1] == b2))
                pos_dest = i;
         }
         printf("\nhgt : (branches) %d %d",pos_source,pos_dest);
         if(pos_source != -1 && pos_dest != -1){
            if(isAValidHGT(SpeciesTree,pos_source,pos_dest) == 1){
               dedans=1;
               if(step != newStep){
                  if(newStep >= 0){
                     imc++;
                  }
                  else{
                     imc=1;
                  }
                  newStep = step;
                  multicheckTab[0].m ++;
                  multicheckTab[imc].nbHgtFound =0;
               }

               printf("\nhgt : (branches) %d %d",pos_source,pos_dest);
               applyHGT(NULL,&SpeciesTree,pos_source,pos_dest);
               AdjustBranchLength(&SpeciesTree,GeneTree,binaireSpecies,1);
               computeCriteria(SpeciesTree.ADD,GeneTree.ADD,SpeciesTree.size,&aCrit,
                     SpeciesTree.LONGUEUR,SpeciesTree.ARETE,GeneTree.LONGUEUR,GeneTree.ARETE);
               printf("\nhgt : %d-%d -> %d-%d",a1,a2,b1,b2);
               printf("\nhgt : CRITERIA RF=%d,LS=%lf,BD=%lf\n",aCrit.RF,aCrit.LS,aCrit.BD);

               multicheckTab[imc].nbHgtFound ++;
               cpt_hgt ++;
               bestHGT[cpt_hgt].source_A = a1;
               bestHGT[cpt_hgt].source_B = a2;
               bestHGT[cpt_hgt].dest_A = b1;
               bestHGT[cpt_hgt].dest_B = b2;
               bestHGT[cpt_hgt].valide = valide;
               bestHGT[cpt_hgt].crit = aCrit;
               bestHGT[cpt_hgt].source = pos_source;
               bestHGT[cpt_hgt].destination = pos_dest;
               bestHGT[cpt_hgt].sequence = imc;
               if((bestHGT[cpt_hgt].source_A == bestHGT[cpt_hgt].dest_A ||
                  (bestHGT[cpt_hgt].source_A == bestHGT[cpt_hgt].dest_B) ||
                  (bestHGT[cpt_hgt].source_B == bestHGT[cpt_hgt].dest_A) ||
                  (bestHGT[cpt_hgt].source_B == bestHGT[cpt_hgt].dest_B)){
                  bestHGT[cpt_hgt].trivial = 1;
                  bestHGT[cpt_hgt].valide = TRIVIAL;
               }
               if(valide == 3){
                  bestHGT[cpt_hgt].valide = valide;
               }

               multicheckTab[imc].LS = aCrit.LS;
               multicheckTab[imc].RF = aCrit.RF;
               multicheckTab[imc].BD = aCrit.BD;
               multicheckTab[imc].rLS = -1;
               multicheckTab[imc].rRF = -1;
               multicheckTab[imc].rBD = -1;
               findListSpecies(&bestHGT[cpt_hgt],DTSpecies,SpeciesTreeCurrent);

               InitCriteria(&aCrit,SpeciesTree.size);
            }
         }
      }
   }

   SpeciesTreeCurrent.ADD=NULL;
   SpeciesTreeCurrent.ARETE=NULL;
   copyInputTree(&SpeciesTreeCurrent,SpeciesTree,1,0);
   AdjustBranchLength(&SpeciesTreeCurrent,GeneTree,binaireSpecies,1);
   AdjustBranchLength(&SpeciesTree,GeneTree,binaireSpecies,1);

   DTGene = (struct DescTree*)malloc((2*GeneTree.size-2-GeneTree.kt+1)*
                                     sizeof(struct DescTree));
   RechercherBipartition(GeneTree.ARETE,GeneTree.ADD,GeneTree.Root,GeneTree.Adjacence,
                         DTGene,GeneTree.size,GeneTree.kt);
```

```
    DTSpecies = (struct DescTree*)malloc((2*SpeciesTree.size-2-SpeciesTree.kt+1)*
                                         sizeof(struct DescTree));
    RechercherBipartition(SpeciesTree.ARETE,SpeciesTree.ADD,SpeciesTree.Root,
                          SpeciesTree.Adjacence,DTSpecies,SpeciesTree.size,SpeciesTree.kt);

    printf("\nhgt : pre-treatment process");
    int tousLesCasSontTraitees = FALSE;
    int *tab_tous_les_sommets = (int*)malloc((SpeciesTree.size+1) * sizeof(int));
    int *tab_sommets_selectionnes = (int*)malloc((SpeciesTree.size+1) * sizeof(int));
    int *tab_branches = (int*)malloc( 4*(GeneTree.size+1) * sizeof(int));
    int nb_branches;
    int temoin_nouveau_cas;
    struct HGT aHGT;

    for(i=1;i<=SpeciesTree.size;i++) tab_tous_les_sommets[i] = 0;
    tab_tous_les_sommets[0] = FALSE;

    while(tousLesCasSontTraitees == FALSE){

       ListeSommets_taille_0(GeneTree.Input,tab_tous_les_sommets,GeneTree.size-1);
       tousLesCasSontTraitees = tab_tous_les_sommets[0];

       if(tousLesCasSontTraitees == FALSE){
          tab_sommets_selectionnes[0] = 0;
          temoin_nouveau_cas=0;
          for(i=1;i<GeneTree.size;i++){
             if(tab_tous_les_sommets[i] == 1){
                if(temoin_nouveau_cas == 0){
                   temoin_nouveau_cas=1;
                }
                tab_tous_les_sommets[i] = 2;
                tab_sommets_selectionnes[0] = tab_sommets_selectionnes[0] + 1;
                tab_sommets_selectionnes[tab_sommets_selectionnes[0]] = i;
             }
          }
       if(tab_sommets_selectionnes[0] > 0){
          ListesBranchesPourHGT(tab_sommets_selectionnes,GeneTree.ARETE,GeneTree.size,
                                DTGene,tab_branches,&nb_branches);
          while(findBestHGT_nombreLimite(DTGene,DTSpecies,tab_branches,nb_branches,
                GeneTree,SpeciesTree,param,&aHGT) > 0){
             applyHGT(SpeciesTree.ADD,&GeneTree,aHGT.source,aHGT.destination);
             AdjustBranchLength(&GeneTree,SpeciesTree,0,1);
             deleteBipartition(DTGene,GeneTree);
             DTGene = (struct DescTree*)malloc((2*GeneTree.size-2-GeneTree.kt+1)*
                                               sizeof(struct DescTree));
             RechercherBipartition(GeneTree.ARETE,GeneTree.ADD,GeneTree.Root,
                                   GeneTree.Adjacence,DTGene,GeneTree.size,GeneTree.kt);
             ListesBranchesPourHGT(tab_sommets_selectionnes,GeneTree.ARETE,GeneTree.size,
                                   DTGene,tab_branches,&nb_branches);
          }
       }
    }
}
free(tab_tous_les_sommets);
free(tab_sommets_selectionnes);
free(tab_branches);

InitCriteria(&aCrit,SpeciesTree.size);
computeCriteria(SpeciesTree.ADD,GeneTree.ADD,SpeciesTree.size,&aCrit,SpeciesTree.LONGUEUR,
                SpeciesTree.ARETE,GeneTree.LONGUEUR,GeneTree.ARETE);

ReduceTree(SpeciesTree,GeneTree,&SpeciesTreeRed,&GeneTreeRed,&aMap,DTSpecies,DTGene,
           binaireSpecies,binaireGene);

tmp = 0;

//============================== DÉTECTION DES TRANSFERTS ==================================

if(strcmp(param.version,"consol")==0){
   printf("\n=============================================");
   printf("\n| CRITERIA VALUES BEFORE THE DETECTION ");
   printf("\n| RF distance  = %2d",aCrit.RF);
   printf("\n| LS criterion = %2.11f",aCrit.LS);
   printf("\n| BD criterion = %2.11f",aCrit.BD);
```

```
    printf("\n==========================================\n");
}

//=============================================================================
//============ RECHERCHE DE TRANSFERTS : scénario multiple  ===================
//=============================================================================
if(strcmp(param.scenario,"multiple")==0){
   cpt_hgt = findAllHGT(SpeciesTreeRed,GeneTreeRed,param,bestHGTRed);
   for(i=1;i<=cpt_hgt;i++){
      expandBestHGT(bestHGTRed[i],&bestHGT[i],aMap,DTSpecies,SpeciesTree);
   }
   sortHGT(bestHGT,cpt_hgt,param);
   printf("\nhgt : scenario multiple");
   if(cpt_hgt > param.nbhgt) cpt_hgt = param.nbhgt;
}
else if(strcmp(param.scenario,"unique")==0){
   printf("\nhgt : scenario unique");
   if( strcmp(param.subtree,"yes") == 0 && strcmp(param.mode,"multicheck")==0 ){

      //=========================================================================
      //======= RECHERCHE DE TRANSFERTS : scénario unique - PLUSIEURS PAR TOUR =======
      //=========================================================================
      printf("\nhgt : start of detection");
      trivial = (SpeciesTree.kt == 0)?0:1;
      while( findBestHGTtab(SpeciesTreeRed,GeneTreeRed,param,bestHGTRed,
                            &nbHgtFound,&trivial,bootstrap) > 0){
         imc++;
         trivial = (SpeciesTree.kt == 0)?0:1;

         printf("\n\n[%d HGT%s]", nbHgtFound,(nbHgtFound > 1)?"s":"");

         if(first==1){
            multicheckTab[0].m ++; //= nombre d'occurences
            multicheckTab[imc].nbHgtFound = nbHgtFound;
         }

         int temoin_zero=-1;
         int cpt_hgt2 = cpt_hgt;
         for(i=0;i<nbHgtFound;i++){
            if(bestHGTRed[i].crit.RF == 0){
               temoin_zero = i;
            }
         }
         printf("\nHGT-DETECTION : cpt_hgt= %d",cpt_hgt);
         for(i=0;i<nbHgtFound;i++){
            cpt_hgt++;

            expandBestHGT(bestHGTRed[i],&bestHGT[cpt_hgt],aMap,DTSpecies,SpeciesTree);
            bestHGT[cpt_hgt].sequence = imc;
            bestHGT[cpt_hgt].trivial = 0;
            bestHGTRed[i].listSource = NULL;
            bestHGTRed[i].listDestination = NULL;
            if((temoin_zero != -1)&&(i!=temoin_zero)){
               bestHGT[cpt_hgt].valide = 0;
               printf("\nHGT-DETECTION : un des transfert met RF=0");
               continue;
            }

            if((bestHGT[cpt_hgt].source_A == bestHGT[cpt_hgt].dest_A) ||
               (bestHGT[cpt_hgt].source_A == bestHGT[cpt_hgt].dest_B) ||
               (bestHGT[cpt_hgt].source_B == bestHGT[cpt_hgt].dest_A) ||
               (bestHGT[cpt_hgt].source_B == bestHGT[cpt_hgt].dest_B)){
               bestHGT[cpt_hgt].trivial = 1;
               bestHGT[cpt_hgt].valide = TRIVIAL;
            }

            if((bestHGT[cpt_hgt].source_A == bestHGT[cpt_hgt].dest_A &&
               SpeciesTree.degre[bestHGT[cpt_hgt].source_A] == 3) ||
               (bestHGT[cpt_hgt].source_A == bestHGT[cpt_hgt].dest_B &&
               SpeciesTree.degre[bestHGT[cpt_hgt].source_A] == 3) ||
               (bestHGT[cpt_hgt].source_B == bestHGT[cpt_hgt].dest_A &&
               SpeciesTree.degre[bestHGT[cpt_hgt].source_B] == 3) ||
               (bestHGT[cpt_hgt].source_B == bestHGT[cpt_hgt].dest_B &&
               SpeciesTree.degre[bestHGT[cpt_hgt].source_B] == 3) ){
```

```
            bestHGT[cpt_hgt].valide = 0;
            printf("\nhgt : useless=> HGT #%d : [%2d--%2d] -> [%2d--%2d]",cpt_hgt,
                                    bestHGT[cpt_hgt].source_A,bestHGT[cpt_hgt].source_B,
                                    bestHGT[cpt_hgt].dest_A,bestHGT[cpt_hgt].dest_B);

            continue;
         }
      }

      initInputTree(&FirstTree2);
      copyInputTree(&FirstTree2,SpeciesTree,0,0);
      cpt_hgt -= nbHgtFound;
      printf("\nHGT-DETECTION : cpt_hgt= %d",cpt_hgt);
      for(i=0;i<nbHgtFound;i++){
         cpt_hgt++;
         if(bestHGT[cpt_hgt].valide > 0){
            SpeciesTree.ADD = NULL;
            SpeciesTree.ARETE = NULL;
            copyInputTree(&SpeciesTree,FirstTree2,0,0);

            applyHGT2(GeneTree.ADD,&SpeciesTree,bestHGT[cpt_hgt].source,
                      bestHGT[cpt_hgt].destination);

            computeCriteria(SpeciesTree.ADD,GeneTree.ADD,SpeciesTree.size,&aCrit,
                    SpeciesTree.LONGUEUR,SpeciesTree.ARETE,GeneTree.LONGUEUR,GeneTree.ARETE);
            bestHGT[cpt_hgt].crit.LS = aCrit.LS;
            bestHGT[cpt_hgt].crit.RF = aCrit.RF;
            bestHGT[cpt_hgt].crit.BD = aCrit.BD;
            printf("\nRF = %d | LS = %1.2lf | BD = %1.2lf",bestHGT[cpt_hgt].crit.RF,
                                    bestHGT[cpt_hgt].crit.LS,bestHGT[cpt_hgt].crit.BD);
            SpeciesTree.ADD = NULL;
            SpeciesTree.ARETE = NULL;
            copyInputTree(&SpeciesTree,FirstTree2,0,0);
            applyHGT2(GeneTree.ADD,&SpeciesTree,bestHGT[cpt_hgt].destination,
                    bestHGT[cpt_hgt].source);

            computeCriteria(SpeciesTree.ADD,GeneTree.ADD,SpeciesTree.size,&aCrit,
                    SpeciesTree.LONGUEUR,SpeciesTree.ARETE,GeneTree.LONGUEUR,GeneTree.ARETE);
            bestHGT[cpt_hgt].crit.rLS = aCrit.LS;
            bestHGT[cpt_hgt].crit.rRF = aCrit.RF;
            bestHGT[cpt_hgt].crit.rBD = aCrit.BD;
            printf("\nrRF = %d | rLS = %1.2lf | rBD = %1.2lf",bestHGT[cpt_hgt].crit.rRF,
                                    bestHGT[cpt_hgt].crit.rLS,bestHGT[cpt_hgt].crit.rBD);
         }
      }

      SpeciesTree.ADD = NULL;
      SpeciesTree.ARETE = NULL;
      copyInputTree(&SpeciesTree,FirstTree2,0,0);
      cpt_hgt -= nbHgtFound;
      printf("\nHGT-DETECTION : cpt_hgt= %d",cpt_hgt);
      for(i=0;i<nbHgtFound;i++){
         cpt_hgt++;
         if(bestHGT[cpt_hgt].valide > 0){
            printf("\nhgt : HGT #%d : [%2d--%2d] -> [%2d--%2d]",cpt_hgt,
                                    bestHGT[cpt_hgt].source_A,bestHGT[cpt_hgt].source_B,
                                    bestHGT[cpt_hgt].dest_A,bestHGT[cpt_hgt].dest_B);
            applyHGT2(GeneTree.ADD,&SpeciesTree,bestHGT[cpt_hgt].source,
                    bestHGT[cpt_hgt].destination);
         }
      }

      computeCriteria(SpeciesTree.ADD,GeneTree.ADD,SpeciesTree.size,&aCrit,
                  SpeciesTree.LONGUEUR,SpeciesTree.ARETE,GeneTree.LONGUEUR,GeneTree.ARETE);

      printf("\n\nCriteria values after this step :");
      printf("\nRF = %d | LS = %1.2lf | BD = %1.2lf\n",aCrit.RF,aCrit.LS,aCrit.BD,aCrit.QD);

      if(first==1){
         multicheckTab[imc].LS = aCrit.LS;
         multicheckTab[imc].RF = aCrit.RF;
         multicheckTab[imc].BD = aCrit.BD;
         multicheckTab[imc].QD = aCrit.QD;
```

```
        }

        if((cpt_hgt >= param.nbhgt)||(aCrit.RF == 0)) break;

        deleteBipartition(DTSpecies,SpeciesTreeCurrent);
        copyInputTree(&SpeciesTreeCurrent,SpeciesTree,1,1);

        DTSpecies = (struct DescTree*)malloc((2*SpeciesTree.size-2-SpeciesTree.kt+1)*
                                             sizeof(struct DescTree));
        RechercherBipartition(SpeciesTree.ARETE,SpeciesTree.ADD,SpeciesTree.Root,
                              SpeciesTree.Adjacence,DTSpecies,SpeciesTree.size,SpeciesTree.kt);
        FreeMemory_InputTreeReduced(&SpeciesTreeRed,SpeciesTreeRed.size);
        FreeMemory_InputTreeReduced(&GeneTreeRed,GeneTreeRed.size);
        initInputTree(&SpeciesTreeRed);
        initInputTree(&GeneTreeRed);
        free(aMap.map);
        free(aMap.gene);
        free(aMap.species);
        ReduceTree(SpeciesTree,GeneTree,&SpeciesTreeRed,&GeneTreeRed,&aMap,DTSpecies,
                                                DTGene,binaireSpecies,binaireGene);

        if(strcmp(param.stepbystep,"yes") == 0) {
           nomorehgt=1;
           break;
        }
     }
     free(aMap.map);
     free(aMap.gene);
     free(aMap.species);
     deleteBipartition(DTSpecies,SpeciesTreeCurrent);

     int retour=0;
     int cpt,nbTours,j;
     printf("\nhgt : multicheckTab[3].nbHgtFound = %d",multicheckTab[3].nbHgtFound);
     printf("\nhgt : Looking for idling hgt");
     retour = DeleteUseLessHGT(cpt_hgt,bestHGT,SpeciesTree,FirstTree);
     printf("(%d)",retour);
     cpt=1;
     if(first==1){
        printf("\nhgt : multicheckTab[0].m=%d : ",multicheckTab[0].m);
        for(i=1;i<=multicheckTab[0].m;i++){
           nbTours = multicheckTab[i].nbHgtFound;
           printf("(%d) ",nbTours);
           for(j=1;j<=nbTours;j++){
              printf("%d ",cpt);
              if(bestHGT[cpt].valide == 0){
                 multicheckTab[i].nbHgtFound --;
              }
              cpt++;
           }
        }
     }
  }
  else{
     //=============================================================================
     //========== RECHERCHE DE TRANSFERTS : scénario unique - UN PAR TOUR ==========
     //=============================================================================
     int initial=1;
     while(findBestHGT(initial,SpeciesTreeRed,GeneTreeRed,param,&bestHGTRed[cpt_hgt+1]) > 0){
        cpt_hgt++;
        expandBestHGT(bestHGTRed[cpt_hgt],&bestHGT[cpt_hgt],aMap,DTSpecies,SpeciesTree);
        bestHGT[cpt_hgt].trivial = 0;
        if((cpt_hgt,bestHGT[cpt_hgt].source_A == bestHGT[cpt_hgt].dest_A) ||
           (cpt_hgt,bestHGT[cpt_hgt].source_A == bestHGT[cpt_hgt].dest_B) ||
           (cpt_hgt,bestHGT[cpt_hgt].source_B == bestHGT[cpt_hgt].dest_A) ||
           (cpt_hgt,bestHGT[cpt_hgt].source_B == bestHGT[cpt_hgt].dest_B)){
           bestHGT[cpt_hgt].trivial = 1;
           bestHGT[cpt_hgt].valide = TRIVIAL;
        }
        if((bestHGT[cpt_hgt].source_A == bestHGT[cpt_hgt].dest_A &&
           SpeciesTree.degre[bestHGT[cpt_hgt].source_A] == 3) ||
           (bestHGT[cpt_hgt].source_A == bestHGT[cpt_hgt].dest_B &&
           SpeciesTree.degre[bestHGT[cpt_hgt].source_A] == 3) ||
           (bestHGT[cpt_hgt].source_B == bestHGT[cpt_hgt].dest_A &&
```

```
            SpeciesTree.degre[bestHGT[cpt_hgt].source_B] == 3) ||
            (bestHGT[cpt_hgt].source_B == bestHGT[cpt_hgt].dest_B &&
            SpeciesTree.degre[bestHGT[cpt_hgt].source_B] == 3) ){
            bestHGT[cpt_hgt].valide = 0;
            continue;
         }
         bestHGTRed[i].listSource = NULL;
         bestHGTRed[i].listDestination = NULL;

         applyHGT(GeneTree.ADD,&SpeciesTree,bestHGT[cpt_hgt].source,
               bestHGT[cpt_hgt].destination);
         AdjustBranchLength(&SpeciesTree,GeneTree,0,1);

         computeCriteria(SpeciesTree.ADD,GeneTree.ADD,SpeciesTree.size,&aCrit,
                     SpeciesTree.LONGUEUR,SpeciesTree.ARETE,GeneTree.LONGUEUR,GeneTree.ARETE);
         loadCriteria(aCrit,&(bestHGT[cpt_hgt]));

         if(strcmp(param.version,"consol")==0){
            printf("\nHGT #%d %d--%d -> %d--%d",cpt_hgt,bestHGT[cpt_hgt].source_A,
                  bestHGT[cpt_hgt].source_B,bestHGT[cpt_hgt].dest_A,bestHGT[cpt_hgt].dest_B);
            printf("\nRF = %d, LS = %lf, BD = %lf \n",aCrit.RF,aCrit.LS,aCrit.BD,aCrit.QD);
         }
         if(bestHGT[cpt_hgt].crit.RF == 0) break;
         if(cpt_hgt >= param.nbhgt) break;

         deleteBipartition(DTSpecies,SpeciesTreeCurrent);
         copyInputTree(&SpeciesTreeCurrent,SpeciesTree,1,1);
         DTSpecies = (struct DescTree*)malloc((2*SpeciesTree.size-2-SpeciesTree.kt+1)*
                                              sizeof(struct DescTree));
         RechercherBipartition(SpeciesTree.ARETE,SpeciesTree.ADD,SpeciesTree.Root,
                           SpeciesTree.Adjacence,DTSpecies,SpeciesTree.size,SpeciesTree.kt);
         free(aMap.map);
         free(aMap.gene);
         free(aMap.species);
         FreeMemory_InputTreeReduced(&SpeciesTreeRed,SpeciesTreeRed.size);
         FreeMemory_InputTreeReduced(&GeneTreeRed,GeneTreeRed.size);
         initInputTree(&SpeciesTreeRed);
         initInputTree(&GeneTreeRed);

         ReduceTree(SpeciesTree,GeneTree,&SpeciesTreeRed,&GeneTreeRed,&aMap,
                 DTSpecies,DTGene,binaireSpecies,binaireGene);
      }

      initial=0;
      while( findBestHGT(initial,SpeciesTreeRed,GeneTreeRed,param,&bestHGTRed[cpt_hgt+1]) > 0){
         cpt_hgt++;
         expandBestHGT(bestHGTRed[cpt_hgt],&bestHGT[cpt_hgt],aMap,DTSpecies,SpeciesTree);
         bestHGT[cpt_hgt].trivial = 0;
         if((cpt_hgt,bestHGT[cpt_hgt].source_A == bestHGT[cpt_hgt].dest_A) ||
            (cpt_hgt,bestHGT[cpt_hgt].source_A == bestHGT[cpt_hgt].dest_B) ||
            (cpt_hgt,bestHGT[cpt_hgt].source_B == bestHGT[cpt_hgt].dest_A) ||
            (cpt_hgt,bestHGT[cpt_hgt].source_B == bestHGT[cpt_hgt].dest_B)){
            bestHGT[cpt_hgt].trivial = 1;
            bestHGT[cpt_hgt].valide = TRIVIAL;
         }
         if((bestHGT[cpt_hgt].source_A == bestHGT[cpt_hgt].dest_A &&
            SpeciesTree.degre[bestHGT[cpt_hgt].source_A] == 3) ||
            (bestHGT[cpt_hgt].source_A == bestHGT[cpt_hgt].dest_B &&
            SpeciesTree.degre[bestHGT[cpt_hgt].source_A] == 3) ||
            (bestHGT[cpt_hgt].source_B == bestHGT[cpt_hgt].dest_A &&
            SpeciesTree.degre[bestHGT[cpt_hgt].source_B] == 3) ||
            (bestHGT[cpt_hgt].source_B == bestHGT[cpt_hgt].dest_B &&
            SpeciesTree.degre[bestHGT[cpt_hgt].source_B] == 3) ){

            bestHGT[cpt_hgt].valide = 0;
            if(strcmp(param.version,"consol")==0){
               printf("\nuseless=> HGT #%d : [%2d--%2d] -> [%2d--%2d]",cpt_hgt,
                                          bestHGT[cpt_hgt].source_A,bestHGT[cpt_hgt].source_B,
                                          bestHGT[cpt_hgt].dest_A,bestHGT[cpt_hgt].dest_B);
            }
            continue;
         }

         bestHGTRed[i].listSource = NULL;
```

```
            bestHGTRed[i].listDestination = NULL;

            applyHGT(GeneTree.ADD,&SpeciesTree,bestHGT[cpt_hgt].source,
                   bestHGT[cpt_hgt].destination);

            AdjustBranchLength(&SpeciesTree,GeneTree,0,1);

            computeCriteria(SpeciesTree.ADD,GeneTree.ADD,SpeciesTree.size,&aCrit,
                         SpeciesTree.LONGUEUR,SpeciesTree.ARETE,GeneTree.LONGUEUR,GeneTree.ARETE);
            loadCriteria(aCrit,&(bestHGT[cpt_hgt]));

            if(strcmp(param.version,"consol")==0){
               printf("\nHGT #%d %d--%d -> %d--%d",cpt_hgt,bestHGT[cpt_hgt].source_A,
                     bestHGT[cpt_hgt].source_B,bestHGT[cpt_hgt].dest_A,bestHGT[cpt_hgt].dest_B);
               printf("\nRF = %d, LS = %lf, BD = %lf\n",aCrit.RF,aCrit.LS,aCrit.BD,aCrit.QD);
            }
            if(bestHGT[cpt_hgt].crit.RF == 0) {
               int retour;
               do{
                  retour = DeleteUseLessHGT(cpt_hgt,bestHGT,SpeciesTree,FirstTree);
               }while(retour > 0);
               break;
            }
            if(cpt_hgt >= param.nbhgt) break;

            deleteBipartition(DTSpecies,SpeciesTreeCurrent);
            copyInputTree(&SpeciesTreeCurrent,SpeciesTree,1,1);
            free(aMap.map);
            free(aMap.gene);
            free(aMap.species);
            DTSpecies = (struct DescTree*)malloc((2*SpeciesTree.size-2-SpeciesTree.kt+1)*
                                                 sizeof(struct DescTree));
            RechercherBipartition(SpeciesTree.ARETE,SpeciesTree.ADD,SpeciesTree.Root,
                                  SpeciesTree.Adjacence,DTSpecies,SpeciesTree.size,SpeciesTree.kt);
            FreeMemory_InputTreeReduced(&SpeciesTreeRed,SpeciesTreeRed.size);
            FreeMemory_InputTreeReduced(&GeneTreeRed,GeneTreeRed.size);
            initInputTree(&SpeciesTreeRed);
            initInputTree(&GeneTreeRed);
            ReduceTree(SpeciesTree,GeneTree,&SpeciesTreeRed,&GeneTreeRed,&aMap,
                    DTSpecies,DTGene,binaireSpecies,binaireGene);
         }
         free(aMap.map);
         free(aMap.gene);
         free(aMap.species);
         deleteBipartition(DTSpecies,SpeciesTreeCurrent);
      }
   }

   //=============================================================================
   //============================ TRAITEMENT DES RÉSULTATS =======================
   //=============================================================================
   printf("\nhgt : formatting the results");
   outHGT = (struct HGT*)malloc(2*param.nbhgt*sizeof(struct HGT));
   nbHGT = formatResult(bestHGT,cpt_hgt,outHGT,FirstTree);
   printHGT(results,results_bouba,param.stepbystep,multicheckTab,param.mode,Rref,
            FirstTree,outHGT,nbHGT,NULL,param.subtree,param.bootmin);

   saveTree(param.outputWeb,FirstTree,outHGT,1,nbHGT,param.subtree,param.scenario,NULL);
   remove(param.noMoreHgtfile);
   if(nomorehgt == 0){
      FILE * out = fopen(param.noMoreHgtfile,"w+");
      fclose(out);
   }

   //=============================================================================
   //============================ LIBERATION DE LA MÉMOIRE =======================
   //=============================================================================
   deleteBipartition(DTGene,GeneTree);
   FreeMemory_InputTreeReduced(&SpeciesTreeRed,SpeciesTreeRed.size);
   FreeMemory_InputTreeReduced(&GeneTreeRed,GeneTreeRed.size);
   FreeMemory_InputTree(&SpeciesTreeCurrent,SpeciesTreeCurrent.size);
   FreeMemory_InputTree(&GeneTree,GeneTree.size);
   FreeMemory_InputTree(&SpeciesTree,SpeciesTree.size);
   if(bootstrap !=1)
```

```
      FreeMemory_InputTree(&FirstTree,FirstTree.size);
   FreeCriteria(&aCrit,SpeciesTree.size);

   for(i=1;i<=cpt_hgt;i++){
      free(bestHGT[i].listSource);
      free(bestHGT[i].listDestination);
   }
   free(bestHGTRed);
   free(bestHGT);

   fclose(results);

   printf("\nhgt : number of HGT(s) found = %d \nhgt : end of computation,
            check the file results.txt for the program output\n",nbHGT);
   exit(nbHGT);
}
```

C.4 Fonction permettant la détection de plusieurs transferts indépendants par itérations

```
//=============================================================================
//= Cette fonction détecte tous les transferts indépendants entre l'arbre d'espèces modifié et
//= l'arbre de gène. Les résultats sont rangés dans le tableau de HGT "aHGT".
//=============================================================================
int findBestHGTtab(struct InputTree SpeciesTree,struct InputTree GeneTree,struct Parameters
param,struct HGT *aHGT, int *nbHgtFound,int *initial,int bootstrap){

   struct InputTree tmpTree;
   struct InputTree tmpTree2;

   int i,j,k,l,first=1,ret=0,trouve;
   int size = SpeciesTree.size;
   int ktSpecies;
   struct CRITERIA aCrit,aCritRef,aCritRef2,aCrit_tmpTree2;
   struct DescTree *DTSpecies,*DTGene;
   int encore = 0;

   initInputTree(&tmpTree);
   initInputTree(&tmpTree2);

   (*nbHgtFound) = 0;

   printf("\n\n== NEW STEP OF DETECTION == [size of the trees (after
                                          reduction)=%d]",SpeciesTree.size);

   //=========================================================
   //= Initialisation de variables contenant les valeurs des
   //= différents critères d'optimisation
   //=========================================================
   InitCriteria(&aCrit,size);
   InitCriteria(&aCritRef,size);
   InitCriteria(&aCritRef2,size);
   InitCriteria(&aCrit_tmpTree2,size);
   computeCriteria(SpeciesTree.ADD,GeneTree.ADD,size,&aCrit,SpeciesTree.LONGUEUR,
                                 SpeciesTree.ARETE,GeneTree.LONGUEUR,GeneTree.ARETE);
   computeCriteria(SpeciesTree.ADD,GeneTree.ADD,size,&aCritRef,SpeciesTree.LONGUEUR,
                                 SpeciesTree.ARETE,GeneTree.LONGUEUR,GeneTree.ARETE);
   computeCriteria(SpeciesTree.ADD,GeneTree.ADD,size,&aCritRef2,SpeciesTree.LONGUEUR,
                                 SpeciesTree.ARETE,GeneTree.LONGUEUR,GeneTree.ARETE);
   loadCriteria(aCrit,&aHGT[0]);


   DTGene = (struct DescTree*)malloc((2*GeneTree.size-2-GeneTree.kt+1)*sizeof(struct DescTree));
   RechercherBipartition(GeneTree.ARETE,GeneTree.ADD,GeneTree.Root,GeneTree.Adjacence,
                         DTGene,GeneTree.size,GeneTree.kt);

   DTSpecies = (struct DescTree*)malloc((2*SpeciesTree.size-2-SpeciesTree.kt+1)*
                                        sizeof(struct DescTree));
   RechercherBipartition(SpeciesTree.ARETE,SpeciesTree.ADD,SpeciesTree.Root,
                         SpeciesTree.Adjacence,DTSpecies,SpeciesTree.size,SpeciesTree.kt);

   int flag2 = 1;
   int flag3;
   int initial3=1;

   //=========================================================
   //= Début de la détection
   //=========================================================

   do{
      for(i=1;i<2*size-3-SpeciesTree.kt;i++)
         for(j=i+1;j<2*size-3-SpeciesTree.kt;j++){

            //=============================================================
            //= Évalution du transfert entre la branche i et la branche j
            //=============================================================
            trouve=0;
            if(isAValidHGT(SpeciesTree,i,j)==1 && i!=j ){
```

```
            copyInputTree(&tmpTree,SpeciesTree,0,0);

            if(strcmp(param.subtree,"yes") == 0){
               if(TestSubTreeConstraint(SpeciesTree,i,j,DTSpecies,DTGene) == 0)  {
                  continue;
                  tmpTree.ADD = NULL;
                  tmpTree.ARETE = NULL;
               }
               flag3=0;
               if(TestSubTreeLeafs(SpeciesTree,i,j,DTSpecies,DTGene) == 1){
                  flag3=1;
               }
            }

            applyHGT(GeneTree.ADD,&tmpTree,i,j);

            if(strcmp(param.criterion,"ls") == 0)
               AdjustBranchLength(&tmpTree,GeneTree,0,1);

            computeCriteria(tmpTree.ADD,GeneTree.ADD,size,&aCrit,tmpTree.LONGUEUR,
                            tmpTree.ARETE,GeneTree.LONGUEUR,GeneTree.ARETE);

            first=1;
            loadCriteria(aCritRef,&aHGT[(*nbHgtFound)]);

            if(((aCritRef.RF-aCrit.RF) == 1) && (*initial==1) && (
              (SpeciesTree.ARETE[2*i-1] == SpeciesTree.ARETE[2*j-1]) ||
              (SpeciesTree.ARETE[2*i-1] == SpeciesTree.ARETE[2*j-2]) ||
              (SpeciesTree.ARETE[2*i-2] == SpeciesTree.ARETE[2*j-2]) ||
              (SpeciesTree.ARETE[2*i-2] == SpeciesTree.ARETE[2*j-1])
              )) {
               UpdateCriterion(&first,param.criterion,aCrit,&aHGT[(*nbHgtFound)],i,j,0);

               aHGT[(*nbHgtFound)].source_A = SpeciesTree.ARETE[2*i-1];
               aHGT[(*nbHgtFound)].source_B = SpeciesTree.ARETE[2*i-2];
               aHGT[(*nbHgtFound)].dest_A = SpeciesTree.ARETE[2*j-1];
               aHGT[(*nbHgtFound)].dest_B = SpeciesTree.ARETE[2*j-2];

               findListSpecies(&aHGT[(*nbHgtFound)],DTSpecies,SpeciesTree);
               trouve=1;
            }
            else if((*initial==0) && ((flag3 == 1) || (initial3 == 0) )){

               if(TestCriterionAndUpdate(&first,param.criterion,aCrit,
                                         &aHGT[(*nbHgtFound)],i,j,0,0) == 1){

                  aHGT[(*nbHgtFound)].source_A = SpeciesTree.ARETE[2*i-1];
                  aHGT[(*nbHgtFound)].source_B = SpeciesTree.ARETE[2*i-2];
                  aHGT[(*nbHgtFound)].dest_A = SpeciesTree.ARETE[2*j-1];
                  aHGT[(*nbHgtFound)].dest_B = SpeciesTree.ARETE[2*j-2];
                  findListSpecies(&aHGT[(*nbHgtFound)],DTSpecies,SpeciesTree);
                  trouve=1;
               }
            }
         }

         //=========================================
         //= Évaluation du transfert inverse
         //=========================================
         if(isAValidHGT(SpeciesTree,j,i)==1 && i!=j ){
            copyInputTree(&tmpTree2,tmpTree,0,0);
            copyInputTree(&tmpTree,SpeciesTree,0,0);

            if(strcmp(param.subtree,"yes") == 0){
               if(TestSubTreeConstraint(SpeciesTree,j,i,DTSpecies,DTGene) == 0) {
                  continue;
                  tmpTree.ADD = NULL;
                  tmpTree.ARETE = NULL;
               }

               if(TestSubTreeLeafs(SpeciesTree,j,i,DTSpecies,DTGene)==1){
                  flag3=2;
                }
            }
```

```
            applyHGT(GeneTree.ADD,&tmpTree,j,i);

            if(strcmp(param.criterion,"ls") == 0)
               AdjustBranchLength(&tmpTree,GeneTree,0,1);
            computeCriteria(tmpTree.ADD,GeneTree.ADD,size,&aCrit,tmpTree.LONGUEUR,
                            tmpTree.ARETE,GeneTree.LONGUEUR,GeneTree.ARETE);
            if(tmpTree2.ADD == NULL){
               aCrit_tmpTree2.RF = (int) INFINI;
            }
            else{
               computeCriteria(tmpTree.ADD,tmpTree2.ADD,size,&aCrit_tmpTree2,
                               tmpTree.LONGUEUR,tmpTree.ARETE,tmpTree2.LONGUEUR,tmpTree2.ARETE);
            }
            int flag=0;
            first=1;
            int flag_bootstrap = 0;
            if(trouve==0)
               loadCriteria(aCritRef,&aHGT[(*nbHgtFound)]);
            else{
               if((bootstrap == 1)&&(aCrit_tmpTree2.RF == 0)&&
                  (aCrit.RF == aHGT[(*nbHgtFound)].crit.RF) ){
                  flag_bootstrap = 1;
               }
               else{
                  aHGT[(*nbHgtFound)].crit.diff_bd =
                                                  fabs(aCrit.BD - aHGT[(*nbHgtFound)].crit.BD);
                  if((flag2 == 1) && (fabs(aCrit.BD - aHGT[(*nbHgtFound)].crit.BD) <= 1)){
                     if( (aHGT[(*nbHgtFound)].crit.RF - aCrit.RF) >= 1 ){
                        trouve = 0;
                        flag = 2;
                     }
                     else{
                        flag = 1;
                     }
                  }
               }
            }

            if(flag != 1){
               if(((aCritRef.RF-aCrit.RF) == 1) && (*initial==1) && (
                 (SpeciesTree.ARETE[2*i-1] == SpeciesTree.ARETE[2*j-1]) ||
                 (SpeciesTree.ARETE[2*i-1] == SpeciesTree.ARETE[2*j-2]) ||
                 (SpeciesTree.ARETE[2*i-2] == SpeciesTree.ARETE[2*j-2]) ||
                 (SpeciesTree.ARETE[2*i-2] == SpeciesTree.ARETE[2*j-2])
               )) {

                  if ((flag_bootstrap == 1) && (rand_bootstrap == 1)){
                     UpdateCriterion(&first,param.criterion,aCrit,
                                     &aHGT[(*nbHgtFound)],j,i,flag_bootstrap);
                     aHGT[(*nbHgtFound)].source_A = SpeciesTree.ARETE[2*j-1];
                     aHGT[(*nbHgtFound)].source_B = SpeciesTree.ARETE[2*j-2];
                     aHGT[(*nbHgtFound)].dest_A = SpeciesTree.ARETE[2*i-1];
                     aHGT[(*nbHgtFound)].dest_B = SpeciesTree.ARETE[2*i-2];
                     findListSpecies(&aHGT[(*nbHgtFound)],DTSpecies,SpeciesTree);
                     trouve=1;
                  }
               }
               else if(((*initial==0) && ((flag3 == 2) || (initial3 == 0))) ||
                        (flag_bootstrap == 1)){
                  if(TestCriterionAndUpdate(&first,param.criterion,aCrit,
                                            &aHGT[(*nbHgtFound)],j,i,flag,flag_bootstrap) == 1){
                     aHGT[(*nbHgtFound)].source_A = SpeciesTree.ARETE[2*j-1];
                     aHGT[(*nbHgtFound)].source_B = SpeciesTree.ARETE[2*j-2];
                     aHGT[(*nbHgtFound)].dest_A = SpeciesTree.ARETE[2*i-1];
                     aHGT[(*nbHgtFound)].dest_B = SpeciesTree.ARETE[2*i-2];
                     findListSpecies(&aHGT[(*nbHgtFound)],DTSpecies,SpeciesTree);
                     trouve=1;
                  }
               }
            }
         }

         if (trouve == 1){
```

```
                (*nbHgtFound)++;
            }
        }
        encore = 0;

        if((*nbHgtFound == 0) && (flag2==1)){
          flag2 = 0;
          encore = 1;
        }

        if((*nbHgtFound == 0) && (initial3==1) && (*initial==0) ){
           initial3 = 0;
           flag2=1;
           encore = 1;
        }


        if((*nbHgtFound == 0) && (*initial==1)){
           (*initial) = 0;
           encore = 1;
           flag2=1;
        }
    }while(encore == 1);

    deleteBipartition(DTSpecies,SpeciesTree);
    deleteBipartition(DTGene,GeneTree);
    FreeCriteria(&aCrit,size);
    FreeCriteria(&aCritRef,size);
    FreeMemory_InputTree(&tmpTree,tmpTree.size);

    return (*nbHgtFound);
}
```

ANNEXE D : ARTICLES PUBLIES ET SOUMIS POUR PUBLICATION DANS LE CADRE DU PROJET DOCTORAL, SPECIFIANT LA CONTRIBUTION D'ALIX BOC

Makarenkov, V., **A. Boc**, C.F. Delwiche, A. B. Diallo, et H. Philippe. (2006). New efficient algorithm for modeling partial and complete gene transfer scenarios. In *Data Science and Classification*, V. Batagelj, H.-H. Bock, A. Ferligoj, et A. Ziberna (Eds.), IFCS, Studies in Classification, Data Analysis and Knowledge Organization, Springer Verlag, pages 341-349.

Contribution de l'étudiant :

- Développement du modèle et de l'algorithme de détection des transferts horizontaux de gènes (transferts complet et partiel – première version du modèle qui a été amélioré par la suite).
- Application de l'algorithme à l'exemple présenté.
- Première formulation de la contrainte de sous-arbres.
- Cet article a été rédigé de façon conjointe.

Nguyen, D., **A. Boc**, Abdoulaye B. Diallo et V. Makarenkov. (2007). Étude de la classification des bactériophages. Actes des 14-emes Rencontres de la Société Francophone de Classification, ENST de Paris, France, pages 161-164.

Contribution de l'étudiant :

- Développement des algorithmes de détections de transferts de gènes adaptés à cette recherche sur l'étude de la classification des bactériophages.

Makarenkov, V., **A. Boc**, et Alpha B. Diallo (2007). La dissimilarité de bipartitions et son utilisation pour détecter les transferts horizontaux de gènes. Actes des 14-emes Rencontres de la Société Francophone de Classification, ENST de Paris, France, pages 90-93.

Contribution de l'étudiant :

- Définition et description de la dissimilarité de bipartitions.
- Comparaison avec les autres mesures (moindres carrés, Robinson et Foulds).
- Utilisation du modèle de contrainte de sous-arbres.
- Cet article a été rédigé de façon conjointe.

Makarenkov, V., **A. Boc**, Alpha B. Diallo et Abdoulaye B. Diallo. (2008). Algorithms for detecting horizontal gene transfers: Theory and practice. In *Data Mining and Mathematical Programming*, P.M. Pardalos et P. Hansen (Eds.), CRM Proceedings and AMS Lecture Notes, volume 45, pages 159-179.

Contribution de l'étudiant :

- Définition et preuve de NP-complétude du problème de recherche des transferts horizontaux partiels en utilisant le modèle des moindres carrés.
- Développement du modèle des transferts horizontaux partiels basé sur l'optimisation par des moindres carrés.
- Développement d'un premier modèle de validation des transferts complets par bootstrap. Simulations effectuées pour ce dernier modèle.
- Cet article a été rédigé de façon conjointe.

Boc, A., H. Philippe et V. Makarenkov. (2010a). Inferring and validating horizontal gene transfer events using bipartition dissimilarity, *Systematic Biology*, volume 59, pages 195-211.

Contribution de l'étudiant :

- Formulation des Théorèmes 1 et 2 (avec des preuves) relatives à la dissimilarité de bipartitions, à la contrainte de sous-arbres et à l'algorithme de détection des transferts de gènes complets.
- Optimisation de l'algorithme de détection des transferts de gènes complets (en temps d'exécution et en précision des résultats).
- Développement d'un modèle incluant trois procédures de validation des transferts horizontaux complets par bootstrap.
- Simulations montrant l'efficacité de l'utilisation de la contrainte de sous-arbres et de la dissimilarité de bipartitions par rapport aux autres méthodes existantes.
- Application de cette dernière version de l'algorithme aux divers jeux de données.
- La rédaction de cet article a été faite de façon conjointe avec mon directeur de thèse.

Boc, A., A-M. Di Sciullo et V. Makarenkov. (2010b). Classification of the Indo-European languages using a phylogenetic network approach. In *Classification as a Tool for Research*, H. Locarek-Junge et C. Weihs (Eds) proceedings of IFCS 2009. Studies in Classification, Data Analysis, and Knowledge Organization, Springer Berlin-Heidelberg-New York, pages 647-655.

Contribution de l'étudiant :

- Modélisation de l'évolution des langues Indo-Européenne prenant en compte les emprunts de mots.
- Adaptation de l'algorithme de détection des transferts de gènes complets pour ce nouveau modèle.

Makarenkov, V., **A. Boc**, J. Xie, P. Peres-Neto, F.-J. Lapointe P. et Legendre. (2010). Weighted bootstrapping: a correction method for assessing the robustness of phylogenetic trees, *BMC Evolutionary Biology*, volume 10:250.

Contribution de l'étudiant :

- Participation à l'élaboration de la procédure algorithmique et à des simulations validant les méthodes proposées.

Boc, A. et V. Makarenkov. (2011) Towards an accurate identification of mosaic genes and partial horizontal gene transfers. *Nucleic Acids Research*, volume 39(21).

Contribution de l'étudiant :

- Développement de l'approche utilisant la fenêtre coulissante pour la détection des transferts de gènes partiels.
- Simulations et application de la nouvelle méthode aux jeux de données présentées.
- La rédaction de cet article a été faite de façon conjointe avec mon directeur de thèse.

New efficient algorithm for modeling partial and complete gene transfer scenarios

Vladimir Makarenkov[1], Alix Boc[1], Charles F. Delwiche[2], Alpha Boubacar Diallo[1], and Hervé Philippe[3]

[1] Département d'informatique, Université du Québec à Montréal, C.P. 8888, Succ. Centre-Ville, Montréal (Québec), H3C 3P8, Canada,
[2] Cell Biology and Molecular Genetics, HJ Patterson Hall, Bldg. 073, University of Maryland at College Park, MD 20742-5815, USA.
[3] Département de biochimie, Faculté de Médecine, Université de Montréal, C.P. 6128, Succ. Centre-ville, Montréal, QC, H3C 3J7, Canada.

Abstract. In this article we describe a new method allowing one to predict and visualize possible horizontal gene transfer events. It relies either on a metric or topological optimization to estimate the probability of a horizontal gene transfer between any pair of edges in a species phylogeny. Species classification will be examined in the framework of the complete and partial gene transfer models.

1 Introduction

Species evolution has long been modeled using only phylogenetic trees, where each species has a unique most recent ancestor and other interspecies relationships, such as those caused by horizontal gene transfers (HGT) or hybridization, cannot be represented (Legendre and Makarenkov (2002)). HGT is a direct transfer of genetic material from one lineage to another. Bacteria and Archaea have sophisticated mechanisms for the acquisition of new genes through HGT, which may have been favored by natural selection as a more rapid mechanism of adaptation than the alteration of gene functions through numerous mutations (Doolittle (1999)). Several attempts to use network-based models to depict horizontal gene transfers can be found (see for example: Page (1994) or Charleston (1998)). Mirkin et al (1995) put forward a tree reconciliation method that combines different gene trees into a unique species phylogeny. Page and Charleston (1998) described a set of evolutionary rules that should be taken into account in HGT models. Tsirigos and Rigoutsos (2005) introduced a novel method for identifying horizontal transfers that relies on a gene's nucleotide composition and obviates the need for knowledge of codon boundaries. Lake and Rivera (2004) showed that the dynamic deletions and insertions of genes that occur during genome evolution, including those introduced by HGT, may be modeled using techniques similar to those used to model nucleotide substitutions (e.g. general Markov models). Moret et al (2004) presented an overview of the network modeling in phylogenetics. In this paper we continue the work started in Makarenkov

et al (2004), where we described an HGT detection algorithm based on the least-squares optimization. To design a detection algorithm which is mathematically and biologically sound we will consider two possible approaches allowing for complete and partial gene transfer scenarios.

2 Two different ways of transferring genes

Two HGT models are considered in this study. The first model, assumes partial gene transfer. In such a model, the original species phylogeny is transformed into a connected and directed network where a pair of species can be linked by several paths (Figure 1a). The second model assumes complete transfer; the species phylogenetic tree is gradually transformed into the gene tree by adding to it an HGT in each step. During this transformation, only tree structures are considered and modified (Figure 1b).

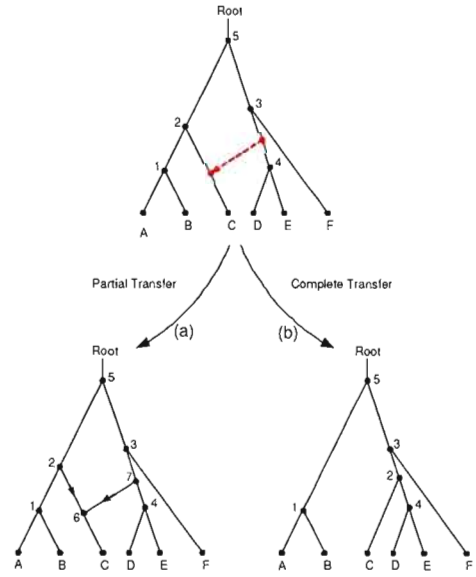


Fig. 1. Two evolutionary models, assuming that either a partial (a) or complete (b) HGT has taken place. In the first case, only a part of the gene is incorporated into the recipient genome and the tree is transformed into a directed network, whereas in the second, the entire donor gene is acquired by the host genome and the species tree is transformed into a different tree.

3 Complete gene transfer model

In this section we discuss the main features of the HGT detection algorithm in the framework of the complete gene transfer model. This model assumes that the entire transferred gene is acquired by the host (Figures 1b). If the homologous gene was present in the host genome, the transferred gene can supplant it. Two optimization criteria will be considered. The first of them is the least-squares (LS) function Q:

$$Q = \sum_i \sum_j (d(i,j) - \delta(i,j))^2, \tag{1}$$

where $d(i,j)$ is the pairwise distance between the leaves i and j in the species phylogenetic tree T and $\delta(i,j)$ the pairwise distance between i and j in the gene tree T_1. The second criterion that can be useful to assess the incongruence between the species and gene phylogenies is the Robinson and Foulds (RF) topological distance (1981). When the RF distance is considered, we can use it as an optimization criterion as follows: All possible transformations (Figure 1b) of the species tree, consisting of transferring one of its subtrees from one edge to another, are evaluated in a way that the RF distance between the transformed species tree T' and the gene tree T_1 is computed. The subtree transfer providing the minimum of the RF distance between T' and T_1 is retained as a solution. Note that the problem asking to find the minimum number of subtree transfer operations necessary to transform one tree into another has been shown to be NP-hard but approximable to within a factor of 3 (Hein et al (1996)).

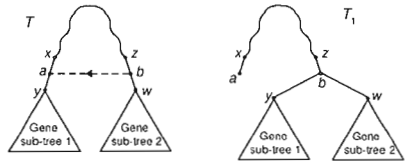


Fig. 2. Timing constraint: the transfer between the edges (z,w) and (x,y) of the species tree T can be allowed if and only if the cluster regrouping both affected subtrees is present in the gene tree T_1.

Several biological rules have to be considered in order to synchronize the way of evolution within a species phylogeny (Page and Charleston (1998)). For instance, transfers between the species of the same lineage must be prohibited. In addition, our algorithm relies on the following timing constraint: The cluster combining the subtrees rooted by the vertices y and w must be present in the gene tree T_1 in order to allow an HGT between the edges (z,w) and (x,y) of the species tree T (Figure 2). Such a constraint enables us,

first, to arrange the topological conflicts between T and T_1 that are due to the transfers between single species or their close ancestors and, second, to identify the transfers that have occurred deeper in the phylogeny. The main steps of the HGT detection algorithm are the following:

Step 0. This step consists of inferring the species and gene phylogenies denoted respectively T and T_1 and labeled according to the same set X of n taxa (e.g. species). Both species and gene trees should be explicitly rooted. If the topologies of T and T_1 are identical, we conclude that HGTs are not required to explain the data. If not, either the RF difference between them can be used as a phylogeny transformation index, or the gene tree T_1 can be mapped into the species tree T fitting by least-squares the edge lengths of T to the pairwise distances in T_1 (see Makarenkov and Leclerc (1999)).

Step 1. The goal of this step is to obtain an ordered list L of all possible gene transfer connections between pairs of edges in T. This list will comprise all different directed connections (i.e. HGTs) between pairs of edges in T except the connections between adjacent edges and those violating the evolutionary constraints. Each entry of L is associated with the value of the gain in fit, computed using either LS function or RF distance, found after the addition of the corresponding HGT connection. The computation of the ordered list L requires $O(n^4)$ operations for a phylogenetic tree with n leaves. The first entry of L is then added to the species tree T.

Steps 2 ... k. In the step k, a new tree topology is examined to determine the next transfer by computing the ordered list L of all possible HGTs. The procedure stops when the RF distance equals 0 or the LS coefficient stops decreasing (ideally dropping to 0). Such a procedure requires $O(kn^4)$ operations to add k HGT edges to a phylogenetic tree with n leaves.

4 Partial gene transfer model

The partial gene transfer model is more general, but also more complex and challenging. It presumes that only a part of the transferred gene has been acquired by the host genome through the process of homologous recombination. Mathematically, this means that the traditional species phylogenetic tree is transformed into a directed evolutionary network (Figure 1a). Figure 3 illustrates the case where the evolutionary distance between the taxa i and j may change after the addition of the edge (b,a) representing a partial gene transfer from b to a.

From a biological point of view, it is relevant to consider that the HGT from b to a can affect the distance between the taxa i and j if and only if a is located on the path between i and the root of the tree; the position of j is assumed to be fixed. Thus, in the network T (Figure 3) the evolutionary distance $dist(i,j)$ between the taxa i and j can be computed as follows:

$$dist(i,j) = (1-\mu)d(i,j) + \mu(d(i,a) + d(j,b)), \qquad (2)$$

where μ indicates the fraction (unknown in advance) of the gene being transferred and d is the distance between the vertices in T before the addition of the HGT edge (b,a). A number of biological rules, not discussed here due to the space limitation, have to be incorporated into this model (see Makarenkov et al (2004) for more details). Here we describe the main features of the network-building algorithm:

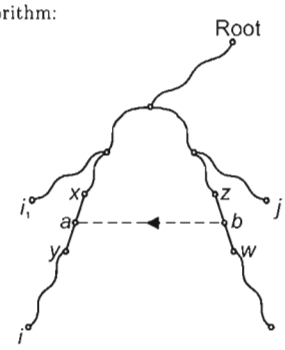


Fig. 3. Evolutionary distance between the taxa i and j can be affected by the addition of the edge (b,a) representing a partial HGT between the edges (z,w) and (x,y). Evolutionary distance between the taxa i_1 and j cannot be affected by the addition of (b,a).

Step 0. This step corresponds to Step 0 defined for the complete gene transfer model. It consists of inferring the species and gene phylogenies denoted respectively T and T_1. Because the classical RF distance is defined only for tree topologies, we use the LS optimization when modeling partial HGT.

Step 1. Assume that a partial HGT between the edges (z,w) and (x,y) (Figure 3) of the species tree T has taken place. The lengths of all edges in T should be reassessed after the addition of (b,a), whereas the length of (b,a) is assumed to be 0. To reassess the edge lengths of T, we have first to make an assumption about the value of the parameter μ (Equation 2) indicating the gene fraction being transferred. This parameter can be estimated either by comparing sequence data corresponding to the subtrees rooted by the vertices y and w or by testing different values of μ in the optimization problem. Fixing this parameter, we reduce to a linear system the system of equations establishing the correspondence between the experimental gene distances and the path-length distances in the HGT network. This system having generally more variables (i.e. edge lengths of T) than equations (i.e. pairwise distances in T; number of equations is always $n(n\text{-}1)/2$ for n taxa) can be solved by approximation in the least-squares sense. All pairs of edges in T can be processed in this way. The HGT connection providing the smallest value of the LS coefficient and satisfying the evolutionary constraints will be selected for the addition to the tree T transforming it into a phylogenetic network.

Steps 2 ... k. In the same way, the best second, third and other HGT edges can be added to T, improving in each step the LS fit of the gene distance. The whole procedure requires $O(kn^5)$ operations to build a reticulated network with k HGT edges starting from a species phylogenetic tree with n leaves.

5 Detecting horizontal transfers of PheRS synthetase

In this section, we examine the evolution of the PheRS protein sequences for 32 species including 24 Bacteria, 6 Archaea, and 2 Eukarya (see Woese et al (2000)). The PheRS phylogenetic tree inferred with PHYML (Guindon and Gascuel (2003)) using G-law correction is shown in Figure 4.

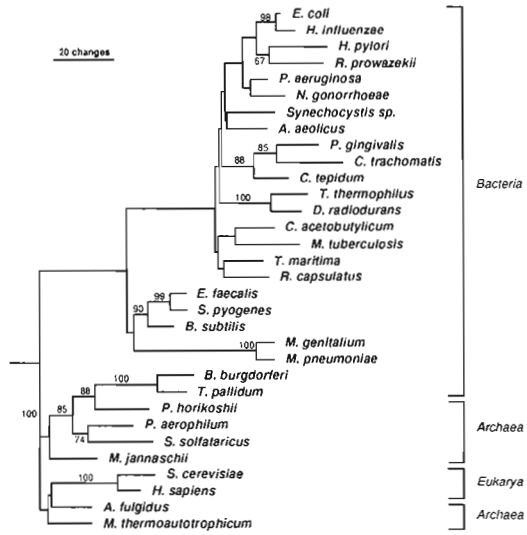


Fig. 4. Phylogenetic tree of PheRS sequences (i.e. gene tree). Protein sequences with 171 bases were considered. Bootstrap scores above 60% are indicated.

This tree is slightly different from the phylogeny obtained by Woese et al (2000, Fig. 2); the biggest difference involves the presence of a new cluster formed by two Eukarya (*H. sapiens* and *S. cerevisiae*) and two Archaea (*A. fulgidus* and *M. thermoautotrophicum*). This 4-species cluster with a low bootstrap support is probably due to the reconstruction artifacts. Otherwise, this tree shows the canonical pattern, the only exception being the spirochete PheRSs (i.e. *B. bugdorferi* and *T. pallidum*). They are of the archaeal, not the bacterial genre, but seem to be specifically related to *P. horokoshii* within

that grouping (Figure 4). The species tree corresponding to the NCBI taxonomic classification was also inferred (Figure 5, undirected lines). The computation of HGTs was done in the framework of the complete gene transfer model. The five transfers with the biggest bootstrap scores are represented.

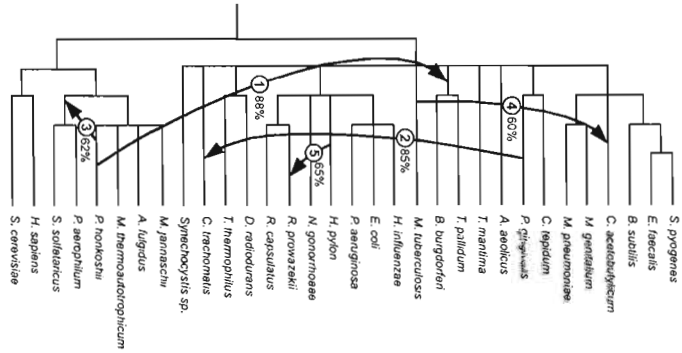


Fig. 5. Species phylogeny corresponding to the NCBI taxonomy for the 32 species in Figure 4. HGTs with bootstrap scores above 60% are depicted by arrows. Numbers on the HGT edges indicate their order of appearance in the transfer scenario.

The bootstrap scores for HGT edges were found fixing the topology of the species tree and resampling the PheRS sequences used to obtain the gene tree. The transfer number 1, having the biggest bootstrap support, 88%, links *P. horokoshii* to the clade of spirochetes. This bootstrap score is the biggest one that could be obtained for this HGT, taking into account the identical 88% score of the corresponding 3-species cluster in the PheRS phylogeny (Figure 4). In total, 14 HGTs, including 5 trivial connections, were found; trivial transfers occur between the adjacent edges. Trivial HGTs are necessary to transform a non-binary tree into a binary one. The non-trivial HGTs with low bootstrap score are most probably due to the tree reconstruction artifacts. For instance, two HGT connections (not shown in Figure 5) linking the cluster of Eukarya to the Archaea (*A. fulgidus* and *M. thermoautotrophicum*) have a low bootstrap support (16% and 32%, respectively). In this example, the solution found with the RF distance was represented. The usage of the LS function leads to the identical scenario differing from that shown in Figure 5 only by the bootstrap scores found for the HGT edges 3 to 5.

6 Conclusion

We described a new distance-based algorithm for the detection and visualization of HGT events. It exploits the discrepancies between the species and gene phylogenies either to map the gene tree into the species tree by least-squares or to compute a topological distance between them and then estimate

the probability of HGT between each pair of edges of the species phylogeny. In this study we considered the complete and partial gene transfer models, implying at each step either the transformation of a species phylogeny into another tree or its transformation into a network structure. The examples of the evolution of the PheRS synthetase considered in the application section showed that the new algorithm can be useful for predicting HGT in real data. In the future, it would be interesting to extend and test this procedure in the framework of the maximum likelihood and maximum parsimony models. The program implementing the new algorithm was included to the T-Rex package (Makarenkov (2001), http://www.trex.uqam.ca).

References

CHARLESTON, M. A. (1998): Jungle: a new solution to the host/parasite phylogeny reconciliation problem. *Math. Bioscience, 149, 191-223.*

DOOLITTLE, W. F. (1999): Phylogenetic classification and the universal tree. *Science, 284, 2124-2129.*

GUINDON, S. and GASCUEL, O. (2003): A simple, fast and accurate algorithm to estimate large phylogenies by maximum likelihood. *Syst. Biol., 52, 696-704.*

LAKE, J. A. and RIVERA, M. C. (2004): Deriving the genomic tree of life in the presence of horizontal gene transfer: conditioned reconstruction. *Mol. Biol. Evol., 21, 681-690.*

LEGENDRE, P. and V. MAKARENKOV. (2002): Reconstruction of biogeographic and evolutionary networks using reticulograms. *Syst. Biol., 51, 199-216.*

MAKARENKOV, V. and LECLERC, B. (1999): An algorithm for the fitting of a tree metric according to a weighted LS criterion. *J. of Classif., 16, 3-26.*

MAKARENKOV, V. (2001): reconstructing and visualizing phylogenetic trees and reticulation networks. *Bioinformatics, 17, 664-668.*

MAKARENKOV, V., BOC, A. and DIALLO, A. B. (2004): Representing lateral gene transfer in species classification. Unique scenario. In: D. Banks, L. House, F. R. McMorris, P. Arabie, and W. Gaul (eds.): *Classification, Clustering and Data Mining Applications.* Springer Verlag, proc. IFCS 2004, Chicago 439-446

MIRKIN, B. G., MUCHNIK, I. and SMITH, T.F. (1995): A Biologically Consistent Model for Comparing Molecular Phylogenies. *J. of Comp. Biol., 2, 493-507.*

MORET, B., NAKHLEH, L., WARNOW, T., LINDER, C., THOLSE, A., PADOLINA, A., SUN, J. and TIMME, R. (2004): Phylogenetic Networks: Modeling, Reconstructibility, Accuracy. *Trans. Comp. Biol. Bioinf., 1, 13-23.*

PAGE, R. D. M. (1994): Maps between trees and cladistic analysis of historical associations among genes, organism and areas. *Syst. Biol., 43, 58-77.*

PAGE, R. D. M. and CHARLESTON, M. A. (1998): Trees within trees: phylogeny and historical associations. *Trends Ecol. Evol., 13, 356-359.*

ROBINSON, D. R. and FOULDS, L. R. (1981): Comparison of phylogenetic trees. *Math. Biosciences, 53, 131-147.*

TSIRIGOS, A. and RIGOUTSOS, I. (2005): A Sensitive, Support-Vector-Machine Method for the Detection of Horizontal Gene Transfers in Viral, Archaeal and Bacterial Genomes. *Nucl. Acids Res., 33, 3699-3707.*

WOESE, C., OLSEN, G., IBBA, M. and SÖLL, D. (2000): Aminoacyl-tRNA synthetases, genetic code, evolut. process. *Micr. Mol. Biol. Rev., 64, 202-236.*

Étude de la classification des bactériophages

D. Nguyen[1], A. Boc[1], A. B. Diallo[1,2] et V. Makarenkov[1]

1. UQAM, CP8888, Succursale Centre-Ville, Montréal (Québec) Canada, H3C 3P8
2. McGill University, 3775 University Street, Montréal (Québec) Canada, H3A 2B4
(Nguyen.Van_Dung.2, Alix.Boc, Diallo.Banire, Makarenkov.Vladimir)@uqam.ca
Banire@mcb.mcgill.ca

Mots clés : classification arborescente, inférence phylogénétique, transfert horizontal de gène.

1. Introduction

L'évolution des bactériophages, qui sont des virus infectant les bactéries et les Archaea, est complexe à cause des mécanismes d'évolution réticulée comprenant le transfert horizontal de gènes (THG) et la recombinaison génétique. Une représentation phylogénétique sous forme de réseau est donc nécessaire pour interpréter l'histoire d'évolution des bactériophages [9]. Par ailleurs, la classification de ces micro-organismes présente intrinsèquement d'autres difficultés dues, d'une part, à la non-conservation de gènes au cours de leur évolution, et d'autre part, à l'hétérogénéité de leurs génomes.

Malgré leur abondance dans la biosphère [6], la classification des bactériophages n'est pas encore complètement établie. Il y a encore beaucoup de possibilités de l'affiner. Dans cet article, nous présentons une plate-forme d'inférence phylogénétique servant à tester les hypothèses sur l'évolution des phages, notamment : a) la reconstruction de l'arbre phylogénétique (i.e. classification) d'espèces de bactériophages, b) la détection et la validation des transferts horizontaux de gènes qui caractérisent leur évolution.

2. Méthodologie

Notre plate-forme d'inférence phylogénétique prend en entrée des données de séquences de protéines et retourne en sortie un *arbre phylogénétique d'espèces* reflétant l'histoire d'évolution des génomes des bactériophages ainsi que des *arbres de gènes* (i.e. des protéines) individuels représentant l'évolution de chacun des gènes considérés. Les différentes statistiques concernant les transferts horizontaux de gènes sont aussi rapportées (voir Figure 1). La méthodologie sous-jacente consiste en trois étapes : préparation de données extraites de la base de données GenBank de NCBI, inférence des arbres d'espèces et de gènes, et détection des THG.

En date de juillet 2006, nous avons recensé sur le site de NCBI, tout en s'assurant de la validité des références sur le site de ICTV (site ayant autorité officielle sur la taxonomie des virus), 163 génomes complets de bactériophages issus de 9 familles différentes dont une avec des annotations partielles (*unclassified*). Les données de séquences de protéines ont été extraites de la banque de données GenBank, en particulier, celles relatives aux VOG – *Viral Orthologous Groups* [1]. Les VOG sont des regroupements prédéfinis de protéines classés selon la fonction protéique à laquelle ils sont associés. Un VOG peut comprendre des séquences de plusieurs espèces différentes. Dans cette étude, 602 regroupements de VOG ont été considérés. L'étude phylogénétique des phages présente un défi double à cause de la grande variabilité à la fois dans la composition génétique et dans la taille des génomes. Le premier défi découle de la grande divergence des séquences de protéines [13]. Le second défi est dû aux tailles de génomes très variables, qui est d'ordre 2 de magnitude (le nombre de gènes codant en protéines varie de 8 à 381), en comparaison aux procaryotes (de ~400 à ~7 000 gènes) ou aux eucaryotes (de ~4 000 à ~60 000), qui sont d'ordre 1 de magnitude [9]. Bien que la meilleure façon de normaliser les génomes de ces micro-organismes en vue d'inférer leur histoire d'évolution reste un débat ouvert [12], la tendance actuelle est de combiner l'étude d'évolution du contenu de gènes et

l'analyse des alignements de chacune des protéines qui se retrouvent dans les génomes de plusieurs phages [9]. Les regroupements des protéines orthologues apportent des données nécessaires pour résoudre la première partie du problème. Reste à trouver le moyen de normaliser correctement.

Le point de départ de notre analyse consiste à estimer les distances entre génomes complets des espèces étudiées. Nous commençons par la construction d'une matrice binaire de présence et d'absence de gènes chez les espèces étudiées. Dans le cas des bactériophages cette matrice comprend 163 lignes (i.e. nombre d'espèces) et 602 colonnes (i.e. nombre de regroupements VOG) contenant des '1' (présence du gène dans le regroupement) et des '0' (absence du gène). Une matrice symétrique de distances inter-génomiques est ensuite calculée. Plusieurs types de distances ont été récemment utilisés pour mesurer la distance entre les génomes : le coefficient de corrélation standard [5], le coefficient de Jaccard [5], le coefficient de Maryland Bridge [12] et la Moyenne Pondérée [3]. Nous avons testé ces différents coefficients. Les résultats obtenus étaient très semblables, compte tenu qu'il n'y ait pas d'ordre *a priori* dans les regroupements VOG (les différences apparaissent seulement au niveau des décimaux).

La matrice de dissimilarité entre les espèces sert ensuite à reconstruire, via l'algorithme Neighbor Joining (NJ) [14], l'arbre phylogénétique d'espèces. Parallèlement à l'algorithme NJ utilisant les distances inter-génomiques, l'approche d'inférence bayésienne, en utilisant le logiciel MrBayes [7], a été examinée. L'avantage de l'approche bayésienne est qu'elle peut traiter directement des caractères morphologiques '0' et '1' sans passer par le calcul de la matrice de distance. Elle suppose une distribution *a posteriori* des topologies d'arbres et utilise les méthodes Markov Chain Monte Carlo (MCMC), pour rechercher dans l'espace d'arbres et inférer la distribution *a posteriori* des topologies.

Pour chacune des deux approches, la validation statistique des topologies d'arbres obtenues a été effectuée. Dans le cas de l'algorithme NJ, le *bootstrap* a été utilisé : a) les données ont été aléatoirement échantillonnées avec remplacements afin de créer de multiple pseudo-données à partir des données d'origine : la matrice binaire d'espèces a été dupliquée en utilisant le programme SeqBoot (inclus dans le package PHYLIP [4]) en 100 copies ; b) avec les copies des matrices binaires, 100 matrices inter-génomiques ont été d'abord calculées, puis servies à reconstruire les arbres d'espèces avec NJ ; c) l'arbre de consensus suivant la règle de majorité étendue (≥50%) a été généré par le logiciel Consens (inclus dans PHYLIP [4]). Dans le cas de MrBayes, 2 millions de générations échantillonnées à toutes les 100 générations, avec 4 chaînes et 2 exécutions indépendantes ont été générées, créant ainsi 20 000 arbres. Un arbre de consensus a été produit à partir des 100 derniers arbres (*burning*=19 900) représentant 10 000 générations stationnaires. Les scores des branches représentent les probabilités *a posteriori* calculées à partir du consensus. En ce qui concerne l'inférence des arbres de gènes, ClustalW [15] a été utilisé pour aligner les séquences appartenant à chacun des 602 VOG. Comme dans le cas de l'arbre d'espèces, NJ et MrBayes ont été appliqués pour reconstruire les arbres de gènes : NJ utilise en entrée les matrices de distances, alors que MrBayes infère l'arbre directement à partir des séquences alignées. Les arbres phylogénétiques d'espèces inférés par NJ et MrBayes ont été très semblables. Toutefois, les scores de probabilités *a posteriori* fournis par MrBayes ont été généralement plus élevés (voir Figure 1) que les scores de bootstrap de NJ. Pour ces données donc l'approche bayésienne se montre plus efficace que celle basée sur les distances. De ce fait, l'approche bayésienne a été retenue pour la suite de nos travaux.

Finalement, la détection des THG a été effectuée, en utilisant le programme HGT Detection du package T-Rex [10], suivant la méthode de réconciliation topologique entre l'arbre de gènes et l'arbre d'espèces [11]. HGT Detection (voir le site www.trex.uqam.ca) prend en entrée un arbre d'espèces et un arbre de gènes pour le même ensemble espèces. Les THG sont ainsi calculés, en indiquant en sortie l'origine et la destination pour chacun des transferts inférés.

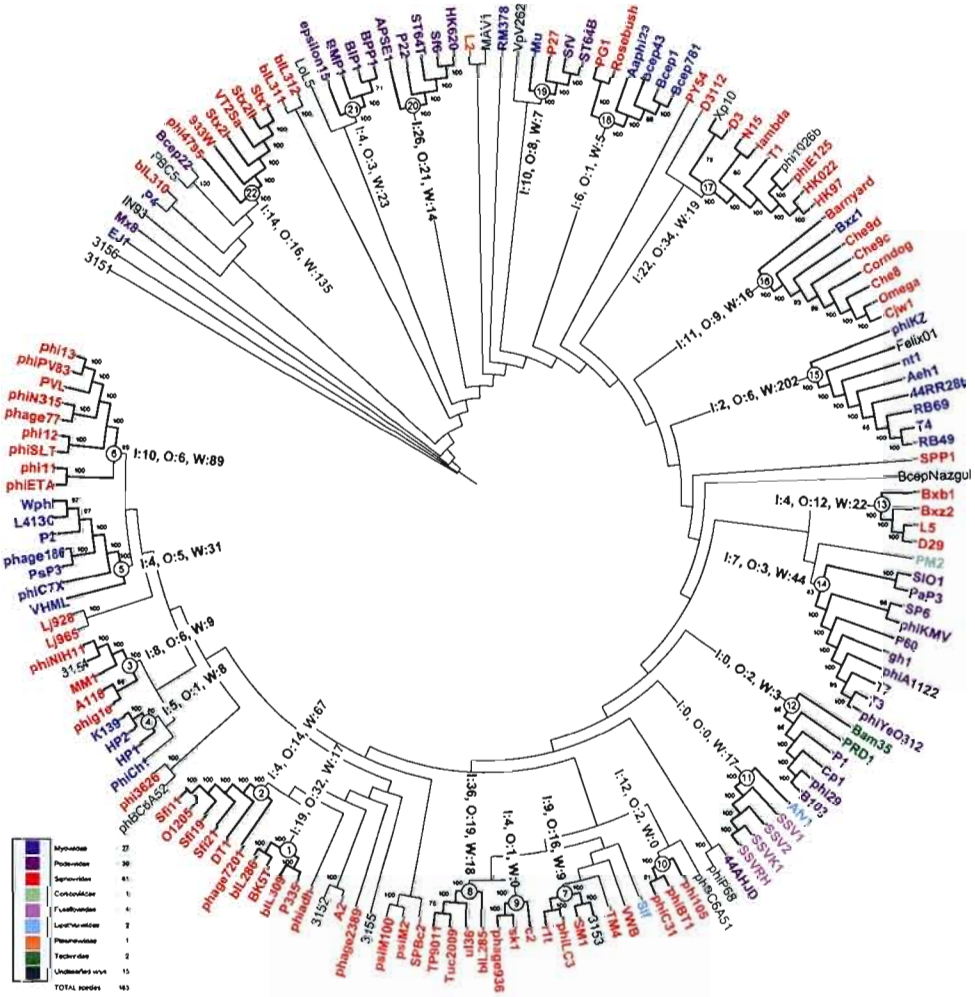


Figure 1 : Arbre phylogénétique d'espèces inféré par MrBayes [7] ; 12 des 22 groupes (représentés par des cercles) identifiés correspondent aux taxonomies de NCBI/ICTV. Pour chaque groupe, I (*In*) signifie le nombre de THG entrant dans le groupe, O (*Out*) le nombre de THG sortant du groupe et W (*Within*) le nombre de THG à l'intérieur de ce groupe.

3. Résultats et discussion

Figure 1 montre l'arbre phylogénétique de bactériophages ainsi que les différentes statistiques obtenues. De manière générale, l'arbre phylogénétique d'espèces, montrant un effet de chaîne, incorpore un grand nombre de signaux phylogénétiques capturés : au total, 122 phages, c-à-d 75% des génomes étudiés, ont été classés dans 22 groupes avec des scores de probabilités *a posteriori* supérieur à 55%. Ces groupes robustes contiennent entre 3 et 10 phages, avec une taille moyenne de clades de 6 espèces. Plusieurs familles d'espèces, 12 sur 22 groupes, référencées par NCBI/ICTV ont été retrouvées par notre analyse : *Siphoviridae* (groupes 1, 2, 6, 8, 9, 10, 13, 22), *Podoviridae* (groupes 14, 20, 21) et *Myoviridae* (groupe 4).

Au niveau des transferts, nous avons calculé les statistiques globales des THG intra (*Within*) et inter (*In/Out*) groupes. Plusieurs points sont remarquables : a) les groupes 2 à 6, 12 à 16, 18,

21 et 22 ont le nombre de transferts intra-groupes supérieur à ceux d'inter-groupes, alors que le reste des groupes a une tendance inverse, à l'exception cependant du groupe 11 qui ne donne ni reçoit de transferts, et des groupes 9 et 10 qui n'ont pas de transferts intra-groupes ; b) les groupes 1, 2, 5, 7, 12, 13, 15, 17 et 22 en donnent plus qu'ils en reçoivent, et inversement pour le reste ; c) les groupes qui en donnent beaucoup plus que la moyenne (informations non représentées sur la figure) sont les suivants : le groupe 1 au groupe 8 (23 transferts), le groupe 17 au groupe 20 (17 transferts) et le groupe 20 au groupe 17 (11 transferts). Les transferts entre les espèces hors groupes n'ont pas été comptabilisés dans cette étude.

Dans le futur proche, nous proposons de compléter cette analyse de l'évolution des bactériophages par une étape supplémentaire. Cette étape consistera à reconstruire les séquences de gènes ancestraux, en utilisant la méthode exposée dans [2], pour chacun des 602 VOG. Cette analyse permettra de déterminer le début de l'histoire évolutive de la fonction protéique associée au VOG en question. Une bonne reconstruction de la séquence protéique ancestrale peut nous aider dans différentes études telles que l'adaptation, le changement de comportement, la divergence fonctionnelle, etc. [8].

4. Bibliographie

[1] Y. Bao, S. Federhen, D. Leipe, V. Pham, S. Resenchuk, M. Rozanov, R. Tatusov, T. Tatusova, "NCBI Genomes Project", *Journal of Virology*, 78:7291-7298. 2004.

[2] A. B. Diallo, V. Makarenkov, M. Blanchette, *"Finding Maximum Likelihood Indel Scenarios"*, *Comparative Genomics*, 171-185. 2006.

[3] B. E. Dutilh, M. A. Huynen, W. J. Bruno, B. Snel, "The consistent phylogenetic signal in genome trees revealed by reducing the impact of noise", *J. Mol. Evol.* 58: 527-539. 2004.

[4] J. Felsenstein, *PHYLIP* (http://evolution.genetics.washington.edu/phylip.html - software download page and software manual) - *PHYLogeny Inference Package*. 2004.

[5] G. Glazko, A. Gordon, A. Mushegian, "The choice of optimal distance measure in genome-wide datasets", *T.P. Biol.* 61, 471–480. 2002.

[6] R. W. Hendrix, "Bacteriophages: evolution of the majority", *T.P. Biol.* 61, 471–480. 2002.

[7] J. P. Huelsenbeck, F. Ronquist, R. Nielsen, J. P. Bollback, "Bayesian inference of phylogeny and its impact on evolutionary biology", *Science* 294:2310–2314. 2001.

[8] N. Krishnan, H. Seligman, C. Stewart, A. Jason de Koning, D Pollock, "Ancestral sequence reconstruction in primate mitochondrial dna: Compositional bias and effect on functional inference", *Molecular Biology and Evolution*. 21 (10), 1871-1883. 2004.

[9] J. Liu, G. Glazko, A. Mushegian, "Protein repertoire of double-stranded DNA bacteriophages", *Virus Research*, 2006 Apr; 117(1):68-80. Epub 2006.

[10] V. Makarenkov, "T-Rex: reconstructing and visualizing phylogenetic trees and reticulation networks", *Bioinformatics* 17, 664-668. 2001.

[11] V. Makarenkov, A. Boc, C. F. Delwiche, A. B. Diallo, H. Philippe (2006). New efficient algorithm for modeling partial and complete gene transfer scenarios. Data Science and Classification, V. Batagelj, H.-H. Bock, A. Ferligoj, and A. Ziberna (Eds.), IFCS 2006, Springer Verlag, pp. 341-349.

[12] B. Mirkin, E. Koonin, "A top-down method for building genome classification trees with linear binary hierarchies", DIMACS series in Discrete Math. & Theor. Computer Sci. 2003.

[13] F. Rohwer, R. Edwards, "The phage proteomic tree: a genome-based taxonomy for phage". J. Bacteriol. 184, 4529–4535. 2002.

[14] N. Saitou, M. Nei, "The Neighbor-Joining method: a new method for reconstructing phylogenetic trees", *Molecular Biology and Evolution*, 4:406-425. 1987.

[15] J. D. Thompson, D. G. Higgins and T. J. Gibson, "CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, positions-specific gap penalties and weight matrix choice", *Nucleic Acids Res.*, 22:4673-4680. 1994.

La dissimilarité de bipartition et son utilisation pour détecter les transferts horizontaux de gènes

Vladimir Makarenkov, Alix Boc et Alpha Boubacar Diallo

Département d'informatique, Université du Québec à Montréal, C.P. 8888, Succ. Centre-Ville, Montréal (Québec), H3C 3P8, Canada
Courriels : (makarenkov.vladimir, boc.alix, diallo.alpha_boubacar)@uqam.ca

Mots clés : dissimilarité, bipartition, arbre phylogénétique, transfert horizontal de gènes.

1 Introduction

Le transfert horizontal de gènes (THG) est un transfert direct du matériel génétique d'une lignée d'un arbre phylogénétique (i.e. arbre additif ou X-arbre [1]) à une autre. Les bactéries et les archéobactéries possèdent des mécanismes sophistiqués qui leur permettent d'acquérir les nouveaux gènes au moyen d'un transfert horizontal. Les trois principaux mécanismes permettant aux espèces de s'échanger des gènes sont la transformation, consistant en l'acquisition de l'ADN directement de l'environnement, la conjugaison, impliquant des plasmides et transposons conjugaux, et la transduction, consistant en des transferts horizontaux par phage [3].

Il existe trois approches pour identifier les gènes qui ont subis des transferts horizontaux. La première consiste à examiner le génome de l'espèce hôte pour voir s'il contient des gènes ayant le contenu en GC ou des motifs de codon atypiques [7]. La deuxième approche consiste à vérifier si le gène étudié, ou sa partie, est présent dans un organisme et absent dans tous les organismes proches. Dans ce cas, il est beaucoup plus probable que ce gène a été introduit dans cet organisme par le THG plutôt que perdu par tous les autres organismes. La troisième approche procède par une comparaison d'un arbre phylogénétique inféré à la base des caractéristiques morphologiques ou à partir d'un gène qui est supposée être résistant aux transferts horizontaux (e.g. souvent on considère 16S rARN ou 23S rARN) et d'un arbre phylogénétique obtenu à partir de la séquence du gène étudié. Le conflit topologique entre ces deux arbres, qui sont appelés respectivement arbre d'espèces et arbre de gène, pourrait être expliqué par les transferts horizontaux. Plusieurs algorithmes permettant d'exploiter les différences topologiques entre les arbres d'espèces et de gène ont été proposés. Mentionnons ici l'algorithme de détection des THG de Hallett et Lagergren [4] qui inscrit des arbres de gène dans un arbre d'espèces et l'algorithme permettant d'identifier simultanément des duplications de gènes, des pertes de gènes, ainsi que des transferts horizontaux de Mirkin et al. [10]. L'article de Moret et al. [11] fait un survol des méthodes utilisées pour détecter des THG de même que d'autres phénomènes d'évolution réticulée. Dans cet article nous décrivons un algorithme polynomial permettant de détecter des THG en utilisant 3 critères d'optimisation différents : les moindres-carrés (MC), la distance topologique de Robinson et Foulds (RF), et la dissimilarité de bipartition. Notons que l'algorithme basé sur les critères MC et RF est celui introduit dans l'article de Makarenkov et al. [9]. La dissimilarité de bipartition sera définie et certaines de ses propriétés seront introduites dans la section suivante. Finalement, les résultats des simulations Monte Carlo effectuées pour comparer les trois critères d'optimisation seront présentés et discutés.

2 Dissimilarité de bipartition et stratégies pour la détection des transferts horizontaux

2.1 *Dissimilarité de bipartition et autres critères d'optimisation*

L'algorithme de détection des THG présenté en détail dans [9] procède par une réconciliation progressive des arbres d'espèces et de gène définis sur le même ensemble de feuilles représentant les espèces étudiées. Ces arbres sont notés T et T', respectivement. Sans perte de généralité nous supposons que les deux arbres sont binaires. À chaque pas de l'algorithme toutes les paires d'arêtes dans T sont testées pour vérifier l'hypothèse qu'un transfert horizontal a eu lieu entre elles. Plus précisément, nous

recherchons le nombre minimum de déplacements de sous-arbres de l'arbre T permettant de le transformer en arbre T'. Évidemment, plusieurs règles d'évolution doivent être incorporées dans le modèle pour le rendre plausible du point de vue biologique. Par exemple, les transferts entre les arêtes appartenant à la même lignée doivent être interdits (voir [8] ou [12] pour plus de détails sur les règles biologiques). Remarquons que le problème de recherche du nombre minimum d'opérations de transferts des sous-arbres nécessaire pour transformer un arbre en un autre a été montré NP-complet (i.e. *Sub-tree transfer problem*, Hein et al. [5]).

Dans ce papier nous présentons 3 critères d'optimisation qui peuvent être incorporés dans un algorithme de détection des THG présenté dans le paragraphe suivant. Le premier critère est celui des moindres carrés Q :

$$Q = \sum_i \sum_j (d(i,j) - \delta(i,j))^2 \qquad (1)$$

où $d(i,j)$ est la distance d'arbre mesurée entre les feuilles i et j dans l'arbre d'espèces T (ou dans l'arbre T_1 obtenu après le premier transfert de sous-arbre dans T) et $\delta(i,j)$ est la distance d'arbre entre les feuilles i et j dans l'arbre de gène T'. Le deuxième critère qui pourrait être utilisé pour estimer la différence entre l'arbre d'espèces et celui de gène est la distance topologique de Robinson et Foulds (RF) [13]. Cette distance est égale au nombre d'opérations élémentaires, consistant en la division et la fusion des nœuds, qui sont nécessaires pour transformer un arbre en un autre. Cette distance est aussi égale au nombre de bipartitions, Buneman [2], qui sont présentes dans un arbre et absentes dans l'autre.

Le troisième critère d'optimisation est la dissimilarité de bipartition que nous introduisons ici. Soient T et T' les arbres phylogénétiques binaires sur le même ensemble d'éléments (i.e. feuilles). Soit **BT** la matrice de bipartition correspondant aux arêtes internes de T et **BT'** la matrice de bipartition de correspondant aux arêtes internes de T'. La dissimilarité de bipartition, bd, entre **BT** et **BT'** est définie comme suit :

$$bd = \Big(\sum_{a \in BT} \underset{b \in BT'}{Min} \left(Min \left(d(a,b); d(a,\bar{b}) \right) \right) + \sum_{b \in BT'} \underset{a \in BT}{Min} \left(Min \left(d(b,a); d(b,\bar{a}) \right) \right) \Big) / 2 \qquad (2)$$

où $d(a,b)$ est la distance de Hamming entre les vecteurs de bipartition a et b, et $\bar{a}$ et $\bar{b}$ sont les compléments de a et de b, respectivement. Une telle mesure pourrait être vue comme une généralisation de la métrique de Robinson et Foulds qui prend en considération seulement des bipartitions identiques.

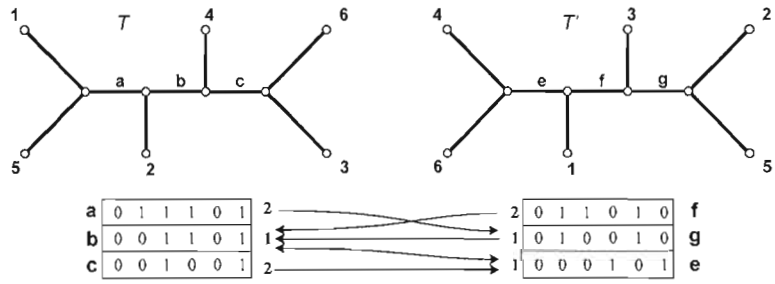


Fig. 1. Les arbres T et T' à 6 feuilles et leur tables de bipartition. Chaque ligne des tables de bipartition correspond à une arête interne. Les flèches indiquent les associations entre les bipartitions dans les deux tables. La valeur en gras à coté de chaque bipartition est la distance de Hamming associée.

Par exemple, la dissimilarité de bipartition bd entre les arbres T et T' à 6 feuilles montrés sur la Figure 1 est calculée comme suit : $bd(T,T') = ((2 + 1 + 2) + (2 + 1 + 1))/2 = 4{,}5$. Ici, le minimum de la distance de Hamming entre la bipartition associée à l'arête a et tous les vecteurs de bipartition dans **BT'** est 2 (la distance entre a et e ou entre a et $\bar{g}$; seulement l'association entre a et e est présentée sur Figure 1). Pour la bipartition associée à l'arête b, cette distance est 1 (la distance entre b et e), et pour la bipartition associée à l'arête c, cette distance est 2 (la distance entre c et e). De la même façon, la

distance de Hamming minimale entre la bipartition associée à f et toutes les bipartitions dans **BT** est 2 (voir la bipartition associée à $\overline{b}$), pour la bipartition associée à g cette distance est 1 (voir la bipartition associée à $\overline{b}$) et pour la bipartition associée à e, elle est aussi 1 (voir la bipartition associée à b).

Cet exemple montre que différents vecteurs de bipartition d'une table de bipartition peuvent être associés au même vecteur de bipartition de l'autre table, e. g. e et g sont associés à b, ainsi que b et c sont associés à e (Figure 1). De plus, une dissimilarité de bipartition n'est pas toujours une métrique. En commençant par des arbres à 5 feuilles nous pouvons exhiber 3 topologies d'arbre pour lesquelles l'inégalité triangulaire n'est pas satisfaite. Une condition suffisante, mais pas nécessaire, pour assurer qu'une dissimilarité de bipartition *bd* est une métrique est la suivante (ici, le symbole $\rightarrow$ désigne l'opération d'association):

Proposition 1. *Soient T_1, T_2 et T_3 des arbres phylogénétiques ayant le même nombre d'arêtes internes et le même ensemble des feuilles. Alors, $bd(T_1, T_2) \leq bd(T_1, T_3) + bd(T_2, T_3)$ si les conditions suivantes sont satisfaites*:

1. *Pour chaque paire de bipartitions a et b de 2 arbres différents: $a \rightarrow b$ signifie que $b \rightarrow a$.*
2. *Pour chaque triplet de bipartitions $a \in T_1$, $b \in T_2$, $c \in T_3$: $a \rightarrow b$ et $b \rightarrow c$ signifie que $a \rightarrow c$.*

Proposition 2. *La valeur d'une dissimilarité de bipartition entre deux arbres phylogénétiques ayant le même ensemble de n feuilles se trouve dans l'intervalle entre 0 et $n(n-3)/2$, si n est paire, et entre 0 et $(n-1)(n-3)/2$, si n est impaire.*

2.2 *Algorithme pour prédire les transferts horizontaux de gènes*

Pas préliminaire

Inférer les arbres phylogénétiques d'espèces et de gène, notés respectivement T et T'. Les feuilles de T et T' sont étiquetées par le même ensemble de n éléments (i.e. d'espèces). Les deux arbres doivent être enracinés. S'il existe dans T et T' des sous-arbres identiques ayant au moins 2 feuilles, réduire la taille du problème en remplaçant dans T et T' les sous-arbres identiques par les mêmes éléments auxiliaires.

Pas 1 ... k

Tester tous les THG possibles entre les paires d'arêtes dans l'arbre T_{k-1} ($T_{k-1} = T$ au Pas 1) à l'exception des transferts entre les arêtes adjacentes et ceux qui violent les contraintes d'évolution (voir [8] ou [12] pour plus de détails). Choisir en tant que THG optimal, le déplacement d'un sous-arbre dans T_{k-1} qui minimise la valeur du critère d'optimisation sélectionné entre l'arbre obtenu après le déplacement de ce sous-arbre et de son greffage sur une nouvelle arête, i.e. l'arbre T_k, et l'arbre de gène T'. Les critères d'optimisation suivants ont été testés : (1) les moindres carrés (MC), (2) la distance topologique de Robinson et Foulds (RF) et (3) la dissimilarité de bipartition (DB). Réduire ensuite la taille du problème en remplaçant des sous-arbres identiques ayant au moins 2 feuilles dans l'arbre d'espèces transformé T_k et l'arbre de gène T'. Dans la liste des THG retrouvés rechercher et éliminer les THG inutiles en utilisant une procédure de programmation dynamique de parcours en arrière. Un transfert inutile est celui dont l'élimination ne change pas la topologie de l'arbre T_k.

Conditions d'arrêt et complexité algorithmique

L'algorithme s'arrête quand le coefficient RF, LS ou BD devient égale à 0 ou quand aucun autre déplacement de sous-arbres n'est possible suite à des contraintes biologiques. Théoriquement, une telle procédure requière $O(kn^4)$ d'opérations pour prédire k transferts dans un arbre phylogénétique à n feuilles. Cependant, due à des réductions inévitables des arbres d'espèces et de gène, la complexité pratique de cet algorithme heuristique est plutôt $O(kn^3)$.

3 Comparaison des trois critères d'optimisation

Cette section présente les performances des 3 stratégies d'optimisation décrites ci-dessus qui peuvent être utilisées pour prédire les transferts horizontaux. Nous montrons ici seulement une partie des résultats obtenus dans nos simulations Monte Carlo. Nous avons généré 100 topologies aléatoires différentes des arbres d'espèces ayant 10, 20, ...et 100 feuilles, respectivement. Chaque topologie a été obtenue en utilisant la procédure de génération d'arbres proposée par Kuhner et Felsenstein [6]. Pour chaque arbre d'espèces, nous avons généré des arbres de gène correspondant aux différents nombres de

transferts tout en respectant les contraintes d'évolution. Le nombre exact de transferts variait de 1 à 10 pour chaque arbre de gène. Nous avons testé les 3 stratégies d'optimisation, MC, RF et DB, présentées dans la section précédente pour mesurer le taux de détection des transferts générés (i.e. *Detection rate* sur la Figure 2a) et le pourcentage des cas quand le nombre exact des transferts générés a été retrouvé (*Same number of HGTs* sur la Figure 2b). Remarquons qu'un transfert correct est celui dont l'emplacement et la direction exacts ont été retrouvés. Pour les deux critères considérés (Figure 2a et b), la stratégie algorithmique basée sur la dissimilarité de bipartition était plus performante que les stratégies basées sur les moindres carrés et la métrique de Robinson et Foulds. Les deux dernières stratégies avaient les tendances similaires, mais celle basée sur la distance RF a toujours fourni des meilleurs résultats que celle basée sur les moindres carrés.

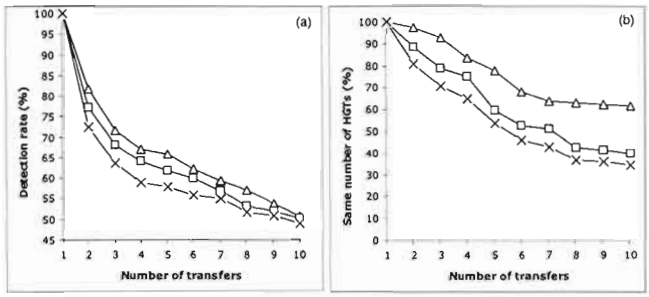


Fig. 2. Pourcentage des cas quand l'algorithme prédit : (a) les transferts corrects ; (b) le nombre exact des transferts - versus le nombre de transferts. Pour chaque point du graphique la moyenne des résultats obtenus pour 100 arbres à 10, 20, ... et 100 feuilles est montrée. Les 3 stratégies comparées sont : la dissimilarité de bipartition (Δ), la distance de Robinson et Foulds (□) et les moindres carrés (×).

4 Références

[1] J.-P. Barthélemy, A. Guénoche, *Les arbres et les représentations des proximités*, Paris, Masson, 1988.

[2] P. Buneman, "The recovery of trees from measures of dissimilarity", *In Mathematics in the Archeological and Historical Sciences, Edinburgh University Press*, 1971, 387–395.

[3] W.F. Doolittle, "Phylogenetic classification and the universal tree", *Science* 284, 1999, 2124-2129.

[4] M. Hallett, J. Lagergren, "Efficient algorithms for lateral gene transfer problems", *In: El-Mabrouk,N., Lengauer,T., Sankoff,D. (eds.): RECOMB, ACM, New-York* 2001, 149-156.

[5] J. Hein, T. Jiang, L. Wang, K. Zhang, "On the complexity of comparing evolutionary trees", *Discr. Appl. Math.* 71, 1996, 153-169.

[6] M. Kuhner, J. Felsenstein, "A simulation comparison of phylogeny algorithms under equal and unequal evolutionary rates", *Mol. Biol. Evol.* 11, 1994, 459-468.

[7] J.G. Lawrence, H. Ochman, "Amelioration of bacterial genomes: rates of change and exchange", *J. Mol. Evol.* 44, 1997, 383-397.

[8] W.P. Maddison, "Gene trees in species trees", *Syst. Biol.* 46, 1997, 523-536.

[9] V. Makarenkov, A. Boc, C. F. Delwiche, A.B. Diallo, H. Philippe, "New efficient algorithm for modeling partial and complete gene transfer scenarios", *Data Science and Classification, V. Batagelj, H.-H. Bock, A. Ferligoj, and A. Ziberna (Eds.), Series: Studies in Classification, Data Analysis, and Knowledge Organization, Springer*, 2006, 341-349.

[10] B. Mirkin, T. I. Fenner, M. Galperin, E. Koonin, "Algorithms for computing parsimonious evolutionary scenarios for genome evolution, the last universal common ancestor and dominance of HGT in the evolution of prokaryotes", *BMC Evol. Biol.* 3, 2003.

[11] B.M.E. Moret, L. Nakhleh, T. Warnow, C.R. Linder, A. Tholse, A. Padolina, J. Sun, R. Timme, "Phylogenetic networks: modeling, reconstructibility, and accuracy", *IEEE/ACM Trans. on Comput. Biol. and Bioinf.* 1, 2004, 13-23.

[12] R.D.M. Page, M.A. Charleston, "Trees within trees: phylogeny and historical associations", *Trends Ecol. Evol.* 13, 1998, 356-359.

[13] D.R. Robinson, L.R. Foulds, "Comparison of phylogenetic trees", *Math. Biosc.* 53, 1981, 131-147.

Centre de Recherches Mathématiques
CRM Proceedings and AMS Lecture Notes Volume
45, 2008, pages 159-179

Algorithms for detecting complete and partial horizontal gene transfers: Theory and practice

Vladimir Makarenkov[1], Alix Boc[1], Alpha Boubacar Diallo[1] and Abdoulaye Baniré Diallo[2]

[1] Département d'informatique, Université du Québec à Montréal, C.P. 8888, Succ. Centre-Ville, Montréal (Québec), Canada, H3C 3P8.

[2] McGill Centre for Bioinformatics and School of Computer Science, McGill University, 3775 University Street, Montréal, Québec, H3A 2B4, Canada.

e-mails: makarenkov.vladimir@uqam.ca, boc.alix@courrier.uqam.ca, diallo.alpha_boubacar@courrier.uqam.ca and banire@mcb.mcgill.ca

Corresponding author: Vladimir Makarenkov (makarenkov.vladimir@uqam.ca).

Abstract. We describe two methods for detecting horizontal gene transfers in the framework of the complete and partial gene transfer models. In case of a complete gene transfer model a new fast backward selection algorithm for predicting horizontal gene transfer events is presented. The latter algorithm can rely either on the metric or on the topological optimization to identify horizontal gene transfers between branches of a given species phylogeny. In case of the topological optimization, we use the well-known Robison and Foulds (RF) topological distance, whereas in case of the metric optimization, the least-squares (LS) criterion is considered. We also formulate and prove the NP-hardness of the partial gene transfer problem. Second, an efficient algorithm for predicting partial transfers, using the Gauss and Seidel optimization, is discussed. We also show how to assess the reliability of a specific gene transfer or a whole gene transfer scenario. In the application section, we apply the new algorithm to detect possible gene transfers occurred during the evolution of the gene *rpl12e*.

INTRODUCTION

Horizontal gene transfer (HGT) is a direct transfer of genetic material from one lineage to another. The understanding that horizontal gene transfer might have played a key role in biological evolution is one of the most fundamental changes in our perception of general aspects of molecular biology in recent years (Doolittle 1999, Legendre 2000, Legendre and Makarenkov 2002). Bacteria and Archaea have sophisticated mechanisms for the acquisition of new genes through HGT which may have been favored by natural selection as a more rapid mechanism of adaptation than the alteration of gene functions through numerous point mutations. If the donor DNA and the recipient chromosome display some homologous sequences, the donor sequences can be stably incorporated into the recipient chromosome by homologous recombination. The three main mechanisms of HGT are the following: transformation, consisting of uptake of naked DNA from the environment; conjugation, which is mediated by conjugal plasmids or conjugal transposons; and transduction, consisting of DNA transfer by phage. These transferring mechanisms can introduce sequences of DNA that display little similarity with the remaining DNA of the recipient cell (Doolittle 1999).

There are a few ways to identify the genes that have been transferred horizontally. First, sequence analysis of the host genome may reveal areas with GC content or codon usage patterns atypical to it (Lawrence and Ochman 1997). Second, if a sequence is found in only one organism and is absent from all other closely related organisms, it is more likely that it has been introduced horizontally into this organism rather than deleted from all the others. Third, the comparison of a morphology-based species tree or a molecular tree based on a molecule that is assumed to be refractory to horizontal gene transfer (e.g. 16S rRNA or 23S rRNA) against a phylogeny of an observed gene may reveal topological conflicts which can be explained by horizontal transfers.

Several attempts to use network-based models to depict horizontal gene transfers can be found (see for example: von Haeseler and Churchill 1993, Page 1994, Charleston 1998, Hallett and Lagergren 2001, or Hallett et al. 2004). A model of horizontal gene transfer that maps gene phylogenies into a species tree has been introduced by Hallett and Lagergren 2001. Mirkin et al. (2003) and Hallett et al. (2004) have developed algorithms allowing for simultaneous identification of gene duplications, gene losses, and horizontal gene transfers. The papers by Moret et al. (2004) and Nakhleh et al. (2005) give an overview of the network modeling in phylogenetics. In a recent paper published in the SFC2004 proceedings, Mirkin (2004) considered some approaches for biologically meaningful mapping of data of individual gene families into an evolutionary species tree. One approach first produces a gene tree, then maps it into the species tree, whereas the other approach first takes the gene phyletic profile, maps it into the species tree and then tunes it into a directed scenario based on the similarity data.

In this article we continue the work started in Boc and Makarenkov (2003), where we described a HGT model based on least-squares, and in Makarenkov et al. (2006), where we showed the difference between complete and partial gene transfer models. First, we describe a polynomial-time HGT algorithm for the detection of complete transfers and test it with respect to the two optimization criteria: Least-squares (LS) and Robinson and Foulds (RF) topological distance. We also suggest how to assess the reliability of

horizontal gene transfers identified by our algorithm. In the application section, we show how the new algorithm predicts transfers of the gene *rpl12e* for the group of 14 Archaea organisms which were originally examined in Matte-Taillez et al. (2002).

ALGORITHMS FOR PREDICTING HORIZONTAL GENE TRANSFERS

2.1 Basic definitions

We start this section with some basic definitions about phylogenetic trees and tree metrics, generally following the terminology of Barthélemy and Guénoche (1988, 1991). The *distance* $\delta(x\ y)$ between two vertices x and y in a phylogenetic (i.e. additive) tree T is defined as the sum of the edge lengths in the unique path linking x and y in T. Such a path is denoted (x,y). A *leaf* is a vertex of degree one.

Definition 1
Let X be a finite set of n taxa. A *dissimilarity* d on X is a non-negative function on $(X\text{x}X)$ such that for any x, y from X:

(1) $d(x,y) = d(y,x)$, and

(2) $d(x,y) = d(y,x) \geq d(x,x) = 0$.

Definition 2
A dissimilarity d on X satisfies the *four-point condition* if for any x, y, z, and w from X:

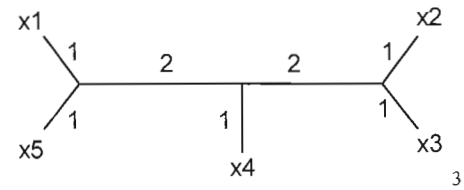
$$d(x,y) + d(z,w) \leq \text{Max}\ \{\ d(x,z) + d(y,w);\ d(x,w) + d(y,z)\ \}.$$

Definition 3
For a finite set X, a **phylogenetic tree** (i.e. an additive tree or a X-tree) is an ordered pair (T, φ) consisting of a tree T, with vertex set V, and a map φ: $X \rightarrow V$ with the property that, for all $x \in X$ with degree at most two, $x \in \varphi(X)$. A phylogenetic tree is **binary** if φ is a bijection from X into the leaf set of T and every interior vertex has degree three.

The main theorem relating the four-point condition and dissimilarity representability by a phylogenetic tree (i.e., phylogeny) is as follows:

Theorem 1 (*Zarestskii, Buneman, Patrinos & Hakimi, Dobson*)
Any dissimilarity satisfying the four-point condition can be represented by a *phylogenetic tree* such that for any x, y from X, $d(x,y)$ is equal to the length of the path linking the leaves x and y in T. This dissimilarity is called a *tree metric*. Furthermore, this tree is unique.
Here is an example of a tree metric on the set X of 5 taxa and the associated phylogenetic tree.

	x2	x3	x4	x5
x1	6	6	4	2
x2		2	4	6
x3			4	6
x4				4



2.2 Optimization criteria

Here we present a fast greedy algorithm for predicting complete horizontal gene transfers. The algorithm for identifying HGTs proceeds by a progressive reconciliation of the given species and gene phylogenetic trees, denoted T and T' respectively. Usually, the species tree T is inferred from the genes that are refractory to horizontal gene transfer and genetic recombination (e.g., 16sRNA sequences). This tree represents the direct or tree-like evolution. The gene tree T' represents the evolution of a given gene which is supposed to undergo horizontal transfers.

At each step of the algorithm, all pairs of branches in T are tested against the hypothesis that a horizontal gene transfer has occurred between them. The considered HGT model assumes that the transferred gene supplants the entire homologous gene of the host or that the homologous gene is simply absent at the host genome. In such a model, the original species phylogenetic tree T is gradually transformed into the gene phylogenetic tree T' through a series of subtree moves (i.e., gene transfers or HGTs). The topology of the gene tree T' is kept fixed. The goal is to find the minimum possible sequence of trees T, T_1, T_2, ..., T' that transforms T into T'. Obviously, a number of necessary biological rules should be taken into account. For instance, the transfers within the same lineage as well as some double-crossing transfers should be prohibited (for more detail, see Maddison 1997, Page and Charleston 1998, or Hallett and Lagergren 2001).

We consider two optimization criteria which can be used at each algorithmic step to select the best HGT. The first optimization criterion that we consider is the *least-squares (LS) function Q*. It is computed as follows:

$$Q = \sum_i \sum_j (d(i,j) - \delta(i,j))^2 \ , \qquad (1)$$

where $d(i,j)$ is the pairwise distance between the leaves i and j in the species tree T (or in the tree T_1 obtained from T after the first subtree move) and $\delta(i,j)$ the pairwise distance between i and j in the gene tree T'. The second criterion that can be useful for assessing discrepancy between the species and gene phylogenies is the *Robinson and Foulds (RF) topological distance*. The RF metric (Robinson and Foulds 1981) is an important and frequently used tool to compare the topologies of phylogenetic trees. This distance is equal to the minimum number of elementary operations, consisting of merging and splitting nodes, necessary to transform one tree into the other. This distance is also the number of bipartitions or Buneman's splits belonging to exactly one of the two trees. When the RF distance is considered, we can use it as an optimization criterion as follows: all possible transformations of the species tree, consisting of transferring one of its subtrees from one branch to another, are evaluated in a way that the RF distance between the transformed species tree T_1 and the gene tree T' is computed. The subtree transfer providing the minimum of the RF distance between T_1 and T' is retained. Note that the problem asking to find the minimum number of subtree transfer operations necessary to transform one tree into another (i.e. also known as *Subtree Transfer Problem*) has been *shown to be NP-hard* (Hein et al. 1996).

2.3 Greedy backward algorithm for predicting complete horizontal gene transfers

In this section we discuss the main features of our algorithm based on the backward selection of horizontal gene transfers. Consider a gene transfer in the species tree T going from b to a and transforming it into the tree T_1 (Figure 1). The following timing constraint is considered (see also Makarenkov et al. 2006): to allow the transfer between the branches (z,w) and (x,y) of the species tree T, the cluster combining the subtrees rooted by the vertices y and w must be present in the gene tree T'. Such a constraint enables us, first, to arrange the topological conflicts between T and T' that are due to the transfers between single species or their close ancestors and, second, to identify the transfers that have occurred deeper in the phylogeny (i.e., closer to the tree root). The usage of this constraint allows the method to follow the order that is opposite to the order of evolution and infer first the most recent HGTs which are easier to detect.

Proposition 1. *If all bipartitions corresponding to the branches of the path (x,z) in the transformed species tree T_1 (Figure 1) can be found in the bipartition table of the gene tree T', then the transfer from b to a, transforming the species tree T into T_1, is a part of a minimum cost HGT scenario transforming T into T'.*

This Proposition can be easily proved by induction on the number of branches of the path (x,z).

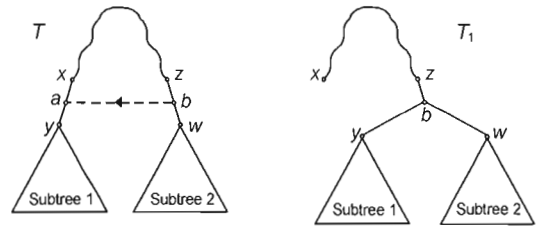


Figure 1. Subtree constraint: the transfer between the branches (z,w) and (x,y) of the species tree T can be allowed if and only if the cluster regrouping both affected subtrees is present in the gene tree; here, a single branch is depicted by a plane line and a path is depicted by a wavy line.

The main steps of the HGT detection algorithm are the following:

Preliminary Step
Infer species and gene phylogenies, denoted respectively T and T', whose leaves are labeled by the same set of n taxa (i.e.,species). Both species and gene trees must be rooted. If there exist identical subtrees with two or more leaves belonging to both T and T', reduce the size of the problem by replacing these subtrees with the same auxiliary taxa in both T and T'.

Steps 1 ... *k*
Test all possible HGTs between pairs of branches in T_{k-1} ($T_{k-1} = T$ at Step 1) except the transfers between adjacent branches and those violating the evolutionary and subtree constraints. If no such a transfer exists, relax the subtree constraint. In our simulations described in the section Simulation study, this relaxation was necessary on average in 1.2% of cases. Search for the transfers satisfying the conditions of Proposition 1. If no such

transfers exist, choose the best HGT with respect to the selected optimization criterion that can be in our case: the least-squares (LS) or the Robinson and Foulds (RF) metric. Reduce the size of the problem by contracting the newly-formed subtree in the transformed species tree T_k and the gene tree T'. In the list of the obtained HGTs, search for and eliminate the idle transfers using a backward procedure. An idle transfer is the transfer whose removal does not change the topology of the tree T_k.

Stopping condition and time complexity

The procedure stops when the LS or RF coefficient equals 0. Such a computation requires $O(kn^4)$ time to generate k transfers in a phylogenetic tree with n leaves. However, because of the progressive size reduction of the species and gene trees, the practical time complexity of this algorithm is rather $O(kn^3)$.

Proposition 2. *If the subtree constraint is not relaxed, the HGT detection algorithm requires at most $n-3$ steps to transform a binary species tree with n leaves into a binary gene tree with the same set of n leaves.*

The proof of this Proposition is based on the fact that the maximum value of the RF distance between two binary trees with n leaves is $2n-6$ and that each subtree transfer satisfying the subtree constraint decreases the value of the RF distance by at least 2.

2.4. Partial gene transfer model

The partial gene transfer model is more general, but also more complex and challenging. It presumes that only a part of the transferred gene has been acquired by the host species through the process of homologous recombination (Makarenkov et al. 2006). This means that the traditional species phylogenetic tree is transformed into a directed phylogenetic network (i.e. a directed connected graph). For example, Denamur et al. (2000) proposed a method to identify gene segments being transferred horizontally. This method was applied to detect partial HGTs of the *mutU* and *mutS* genes within *E. coli* evolutionary trees. Because many analyses are now directed at understanding the evolution of complete genomes, the partial gene transfer model could be also useful if one wanted to model the transfer of a portion of a genome.

In a phylogenetic tree, there is always a unique path connecting a pair of nodes. Adding to it a HGT branch creates an extra path between certain nodes. Figure 2 illustrates the case where the evolutionary distance between the taxa i and j can be affected by the addition of the HGT branch (b,a) representing partial gene transfer from b to a. It is relevant to assume that the HGT from b to a can affect the evolutionary distance between the taxa i and j if and only if the destination point a is located on the path between i and the root of the tree; the position of j is fixed. Thus, in the reticulate phylogeny T in Figure 2 the evolutionary distance $d_1(i,j)$ between the taxa i and j can be computed as follows:

$$d_1(i,j) = (1 - \alpha)\, d(i,j) + \alpha\, (d(i,a) + d(j,b)), \qquad (2)$$

where α indicates the fraction, unknown in advance, of the transferred gene and d is the internode distance in the species tree before the addition of the HGT branch (b,a).

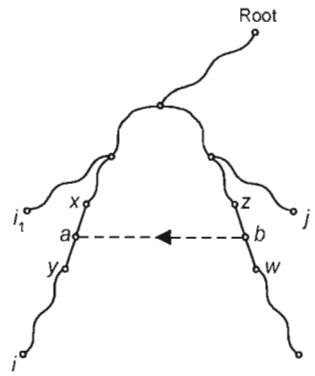


Figure 2. Evolutionary distance between the taxa i and j can be allowed to change after the addition of the branch (b,a) representing a partial HGT between the branches (z,w) and (x,y). Evolutionary distance between the taxa i_1 and j must not be affected by the addition of (b,a).

On the contrary, the distance between the taxa i_1 and j (Figure 2) must not be affected by the addition of (b,a). Figure 3 illustrates the other cases where the addition of a HGT branch must not affect the length of the evolutionary path between i and j.

The least-squares loss function Q to be minimized with the *unknown vector of edge lengths* $\boldsymbol{l}$ in T and the *unknown fraction of the transferred gene* α is as follows:

$$Q(L,\alpha) = \sum_{ij\in S}((1-\alpha)\sum_{k\in path(ij)} l_{ij}^k + \alpha(\sum_{k\in path(ia)} l_{ia}^k + \sum_{k\in path(jb)} l_{jb}^k) - \delta(i,j))^2 + \sum_{ij\notin S}(\sum_{k\in path(ij)} l_{ij}^k - \delta(i,j))^2 \rightarrow min, \quad (3)$$

where $\delta(i,j)$ is the given gene dissimilarity between i and j; l_{ij}^k is the length of the branch k of the path (ij) in T; α is the fraction of the transferred gene ($0 \le \alpha \le 1$); and S is the set of pairs of taxa $\{ij\}$ such that the transfer (ba) can affect the evolutionary distance between them.

To show the NP-hardness of the least-squares optimization in the context of the partial gene transfer the following problem can be stated:

Given: *Species phylogenetic tree T (with the associated tree metric d on the set of taxa X), gene dissimilarity* δ *on X, and a fixed non-negative value* ε.

Problem: *Find the minimum number of partial gene transfers k such that:*

$$Q = \sum_i \sum_j (d_k(i,j) - \delta(i,j))^2 \le \varepsilon, \quad (4)$$

where $d_k(i,j)$ *is the network distance between i and j, computed using Formulae 2 and 3, in the phylogenetic network* T_k *obtained from T after the addition of k partial gene transfers.*

Theorem 2. *The minimum number of partial gene transfer problem (MNPGT problem) is NP-hard.*

The proof of this Theorem is based on a polynomial-time reduction from the *Subtree Transfer Problem* (STR problem) that consists of finding the minimum number of complete gene transfers to transform a given species tree T into a given gene tree T'. The STR problem is identical to the problem of adding to T the minimum number of complete gene transfers such that $Q = \sum_i \sum_j (d_k(i,j) - \delta(i,j))^2 \leq 0$ (i.e., the case of $\varepsilon = 0$ is considered), where $d_k(i,j)$ is the pairwise distance between i and j in the phylogenetic tree (i.e., a particular case of a phylogenetic network). Here, the tree T_k is obtained from T after the addition of k complete gene transfers (i.e., a particular case of a partial transfer) and $\delta(i,j)$ is the given tree metric associated with T'.

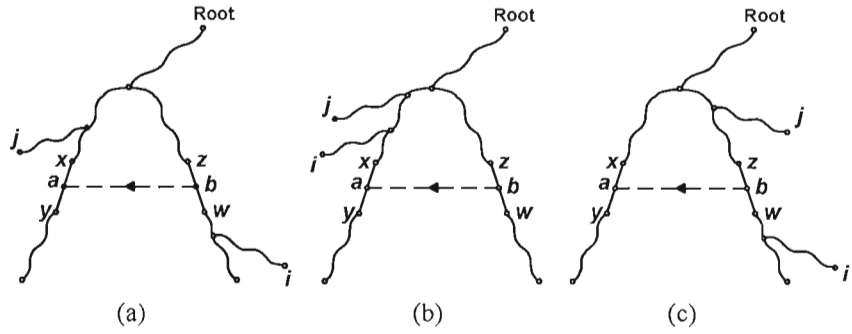


Figure 3 (a-c). Three situations when the evolutionary distance between the taxa i and j must not be affected by the addition of the new branch (b,a) representing a partial HGT between the branches (z,w) and (x,y). Path between the taxa i and j cannot to go through the branch (b,a).

Several important timing constraints have to be incorporated into this model, in addition to those taken into account in the complete HTS model, to identify the interactions between HGTs that are not intelligible from an evolutionary point of view. Some of these constraints, but not all of them, were initially pointed out by Page and Charleston (1998a and b). For instance, double-crossing transfers between two lineages (Figures 4a and b) must be forbidden. In this case, the HGT events affect the ancestor of the species from the previous transfer. Making the source and destination lineages contemporaneous for one HGT makes the other transfer impossible (Figure 4).

Note that the rule illustrated in Figure 4a is automatically taken into account in the complete gene transfer model, where its violation would be equivalent to the violation of the same lineage constraint (see Page and Charleston 1998). For instance (Figure 4a), the HGT from (z,w) to (x,y) cannot be followed by the transfer from (z_1,w_1) to (x_1,y_1) because after the first HGT the branches (z_1,w_1) and (x_1,y_1) will be located on the same lineage (Lineage 2). We also identify two cases, where the evolutionary distance between the taxa i and j can be affected by multiple transfers (Figures 5a and b); and, two cases, where this distance must not be affected by them (Figures 5c and d). Failure to

take these constraints into account can result in postulating transfers that are mutually incompatible.

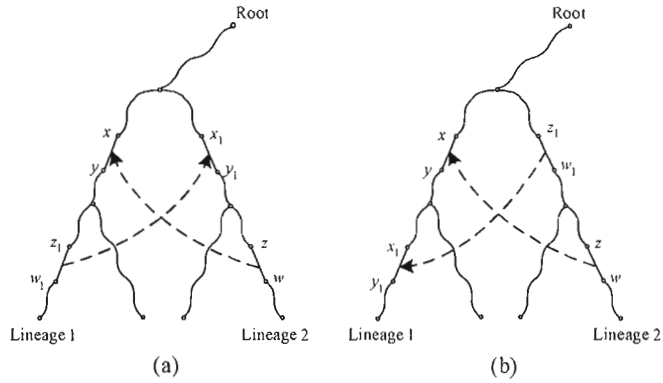


Figure 4. Transfers between two lineages crossing in such ways must be prohibited.

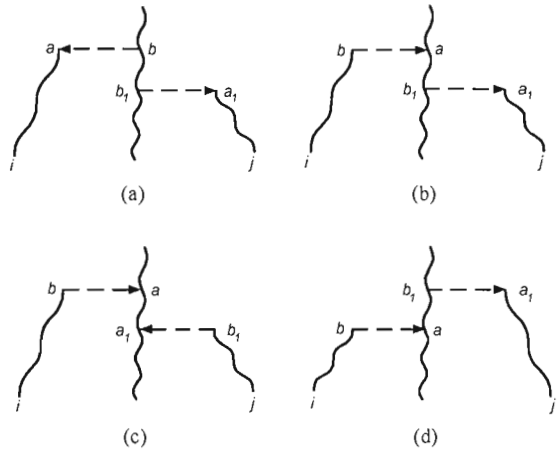


Figure 5. Cases (a) and (b): evolutionary path between the taxa i and j can go through both HGT branches (b,a) and (b_1,a_1). Cases (c) and (d): evolutionary path between the taxa i and j cannot go through both HGT branches (b,a) and (b_1,a_1).

Assume that a partial gene transfer between the branches (z,w) and (x,y) (i.e., from b to a in Figure 2) of the species tree T has taken place. The lengths of all branches in T are reassessed in the least-squares sense after the addition of (b,a), whereas the length of (b,a) is assumed to be 0. To reassess the branch lengths of T, we have first to make an assumption about the value of the parameter α (Equation 2), indicating the gene fraction being transferred. This parameter can be estimated either by comparing sequence data corresponding to the subtrees rooted by the vertices y and w, or different values of α can be tested in the optimization problem.

Fixing the parameter α, we reduce to a linear system the system of equations establishing the correspondence between the experimental gene distances and the path-length distances in the HGT network. This system having generally more variables (i.e. branch lengths of T) than equations (i.e. pairwise distances in T; the number of equations is always $n(n\text{-}1)/2$ for n taxa) can be solved by approximation in the least-squares sense. Let us now show how the approximation problem can be stated and efficiently solved.

Let $\mathbf{A}_\alpha$ be the matrix of dimension $n(n\text{-}1)/2 \times m$, each row of which is associated with one pair of taxa of X, where n is the number of taxa and m is a number of edges in T. The value $a_{ij,e}$ of this matrix corresponding to the pair of taxa ij and the edge e is equal either to 1, or to α, or to $1\text{-}\alpha$ if the edge e is in the path (ij) in T, and is equal to 0 if not. Let ℓ be the vector of edge lengths of m elements and $\mathbf{d}$ be given vector of gene distances of $n(n\text{-}1)/2$ elements.

Fixing the value of α (e.g., values 0, 0.1, 0.2, ..., and 1.0 can be tested in turn), we obtain a linear system of $n(n\text{-}1)/2$ equations with m unknowns: $\mathbf{A}_\alpha \times \ell = \mathbf{d}$.
When $n \geq 4$, this system has more equations than unknowns. It can be solved by approximation in the least-squares sense:

$$(\mathbf{A}_\alpha \times \ell - \mathbf{d})^2 \rightarrow \min. \tag{4}$$

After taking the gradient we have:

$$\mathbf{A}'_\alpha \times (\mathbf{A}_\alpha \times l - \delta) = 0 \tag{5}$$

Following algebraic manipulations, we obtain:

$$\mathbf{A}'_\alpha \times \mathbf{A}_\alpha \times l = \mathbf{A}'_\alpha \times \delta \tag{6}$$

Thus, we have: $\mathbf{B} \times \ell = \mathbf{c}$, where $\mathbf{B}$ is a $(m \times m)$ matrix, and $\mathbf{c}$ is a vector with m components.

Following Barthélemy and Guénoche (1988) and Makarenkov and Leclerc (1999), we apply a slightly modified Gauss-Seidel method to solve the above system. The method consists of decomposing $\mathbf{B}$ into its diagonal (Δ), its strictly upper triangular component ($-\mathbf{F}$), and its strictly lower triangular component ($-\mathbf{E}$):

$$\mathbf{B} = \begin{pmatrix} b_{11} & b_{12} & \cdots & b_{1m} \\ b_{21} & b_{22} & \cdots & b_{2m} \\ \cdots & \cdots & \cdots & \cdots \\ b_{m1} & b_{m2} & \cdots & b_{mm} \end{pmatrix} = \begin{pmatrix} & & -\mathbf{F} \\ & \Delta & \\ -\mathbf{E} & & \end{pmatrix} = \Delta - \mathbf{E} - \mathbf{F}. \tag{7}$$

Then, we apply the iterative procedure:

$$\Delta \times \ell^{(k+1)} = \mathbf{E} \times \ell^{(k+1)} + \mathbf{F} \times \ell^{(k)} + \mathbf{c}, \tag{8}$$

which allows us to compute gradually the components of the vector $\ell(j)^{(k+1)}$, corresponding to the edge lengths at the k+1-th iteration, from those of $\ell(j)^{k}$. If the computed value of $\ell(j)^{(k+1)}$ is negative, it is replaced with the value 0. This operation is equivalent to the projection on the cone $\mathbf{L} \geq 0$, which ensures an appropriate solution.

The exact equation used in this method is the following for all $j = 1, 2, \ldots, m$:

$$\ell(j)^{(k+1)} = (- (\sum_{j+1 \leq i \leq m} b_{ij} \ell(j)^{(k)}) - (\sum_{1 \leq i \leq j-1} b_{ij} \ell(j)^{(k+1)}) + c_j) / b_{jj}. \tag{9}$$

Thus, the main steps of the partial gene transfer algorithm can be stated as follows:

Preliminary Step
This step corresponds to the preliminary step discussed in the context of the complete gene transfer model. It consists of inferring the species and gene phylogenies denoted respectively T and T' whose leaves are labeled by the same set X of n taxa. Because the classical Robinson and Foulds distance is defined only for tree topologies, we use the least-squares as a unique optimization criterion when modeling partial HGTs.

Step 1. Test all connections between pairs of branches in the species tree $\boldsymbol{T}$. For each HGT connexion satisfying evolutionary constraints, carry out the following optimization:

- **a)** Fix the value of the fraction of the gene being transferred α (e.g., one can try in turn the values of 0, 0.1, 0.2, ..., and 1.0). Compute using the Gauss-Seidel method the optimal lengths **l** of the edges in the species tree (or network, starting from Step 2) $\boldsymbol{T}$.
- **b)** Go back to the original equation system: $\mathbf{A}_a \times \boldsymbol{l} = \mathbf{d}$. Fix the values of the vector $\boldsymbol{l}$ found using the Gauss-Seidel method and solve this problem by least-squares considering as unknown the parameter α.
- **c)** Then, fix the optimal value of α found and repeat the computation until both unknown parameters $\boldsymbol{l}$ and α converge to a certain solution.

All eligible pairs of branches in T can be processed in this way. The HGT connection providing the smallest value of the LS coefficient Q and satisfying the defined evolutionary constraints should be selected for the addition to the species tree T, transforming it into a phylogenetic network.

Steps 2...*k*. Run the algorithm until a fixed number k of partial gene transfers is found and added to T or the value of the LS criterion Q is lower than a pre-established threshold ε.
Time complexity of this algorithm is $O(kn^5)$ to add k partial horizontal gene transfers to the species tree with n leaves.

2.5 Bootstrap validation of horizontal gene transfers

Bootstrap analysis can be used to place confidence intervals on internal branches of evolutionary trees (Felsenstein 1985). We designed a bootstrap validation procedure for computing the bootstrap scores either for a specific gene transfer or a whole gene transfer scenario. The following strategy was adopted to assess the reliability of obtained HGTs. Because we are mostly interested in the evolution of a given gene or a group of genes, the sequences used to build the species tree are not resampled. The species tree is taken as an *a priori* assumption of the method and held constant. The sequence data used to build the gene tree are drawn with replacement in order to create a series of pseudo-replicates. The HGT detection algorithm is then carried out on the bootstrapped pseudo-replicates. Thus, for all HGT branches appearing in the original scenario, we verify if they appear in the obtained transfer scenarios, using as input the original species tree and the gene tree inferred from the sets of pseudo-replicates. It is worth noting that among resampled datasets only those that give rise to a gene phylogenetic tree such that it contains the root branch separating this tree into exactly the same bipartition sets as the root branch of the original gene tree does, are eligible for the HGT bootstrap analysis.

SIMULATION STUDY

A Monte Carlo study was conducted to test the ability of the new method to recover correct gene transfers. In the framework of *the complete HGT model only* we examined how the detection procedure performed depending on the model of sequence evolution, number of observed species, and sequence length. The results illustrated in Figures 6 and 7, and reported in Tables 1 and 2 (see Appendix) were obtained from simulations carried out with random binary phylogenetic trees with 8, 16, 24, 32, 48, and 64 leaves, whereas the sequence length varied from 125 to 1000 sites. The simulation procedure consisted of the five basic steps described below:

1. A true tree topology, denoted T, was obtained using the random tree generation procedure proposed by Kuhner and Felsenstein (1994). The branch lengths of T were computed using an exponential distribution. Following the approach of Guindon and Gascuel (2002), we added some noise to the branches of the true phylogenies to create a deviation from the molecular clock hypothesis. All the branch lengths of T were multiplied by $1+ax$, where the variable x was obtained from a standard exponential distribution ($P(x>k) = \exp(-k)$), where the constant a was a tuning factor for the deviation intensity. Following Guindon and Gascuel (2002), a was fixed to 0.8. The random trees generated by this procedure are chosen to have the depth of $O(\log(n))$, where n is the number of species (i.e. number of leaves in a binary phylogenetic tree).

2. Each random phylogeny was then submitted to the SeqGen program (Rambaut and Grassly 1996) to simulate sequence evolution along its branches according to the Jukes and Cantor (1969), Kimura 2-parameter (1980), and Jin-Nei Gamma (1990) models.

3. To assess the quality of HGT detection by the new method, we developed a simulation program using the results of SeqGen. For each considered rooted tree, viewed as an organismal phylogeny, our program created one random horizontal gene transfer that respected the evolutionary constraints discussed in the algorithmic section. During this operation, the program regenerated the DNA sequences for each tree node located in the subtree affected by the HGT. As the simulations were carried out for the complete gene transfer model, the HGT destination sequence was set identical to the source sequence and the new sequences were regenerated from it according to the selected evolutionary model.

4. The sequence to distance transformation corresponding to the considered model of evolution was then applied to the DNA sequences associated with the leaves of the phylogeny affected by the gene transfer. The NJ method (Saitou and Nei 1987) was used to infer the gene trees from the obtained distance matrix. The topology of the organismal phylogeny (i.e. true tree T) was supposed to be known.

5. The HGT detection method was then carried out to infer the transfer. The experiments were conducted using the procedures based on the RF and LS optimization. The simulations were carried out for 500 random rooted phylogenies with 8 and 16 leaves and 100 random rooted phylogenies with 24 to 64 leaves.

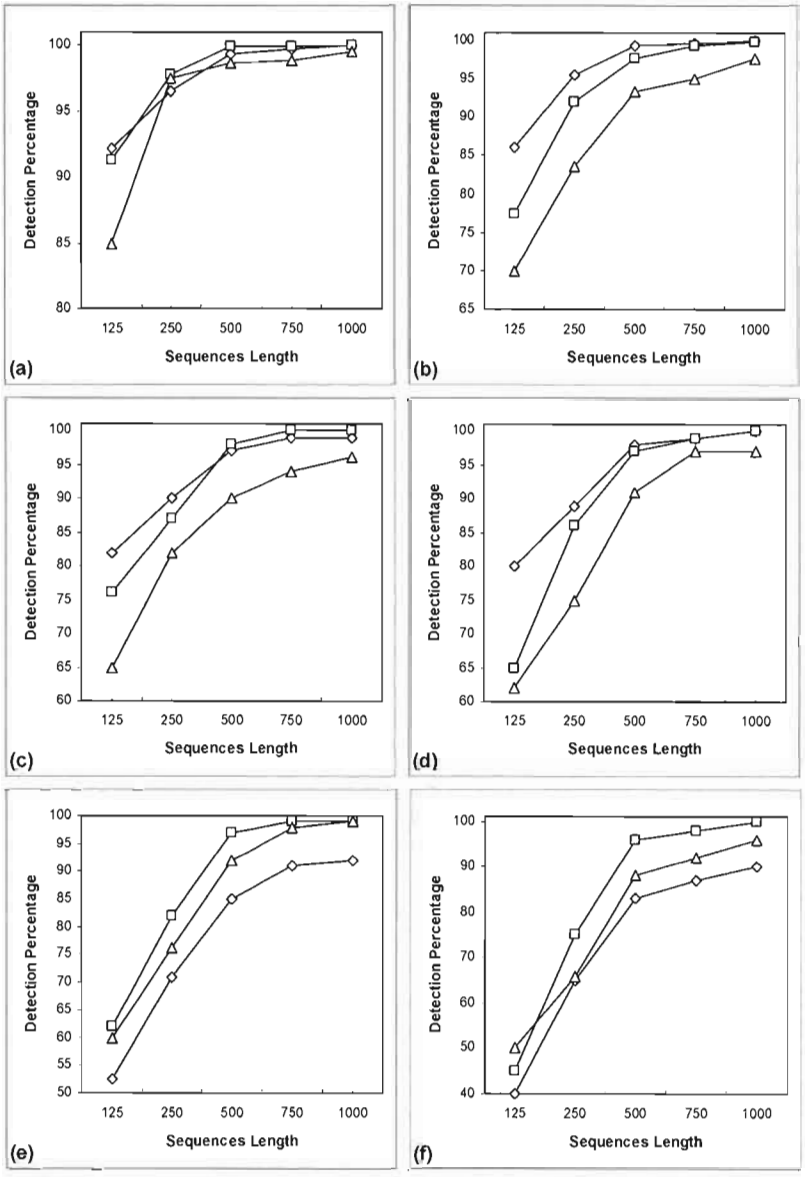


Figure 6. HGT detection rates obtained for random phylogenies with 8 to 64 leaves (8-a, 16-b, 24-b, 32-d, 48-e, 64-f) using the RF topological distance for optimization. Jukes and Cantor (◊), Kimura 2-parameter (□), and Jin-Nei Gamma (Δ) models were used for the tree generation.

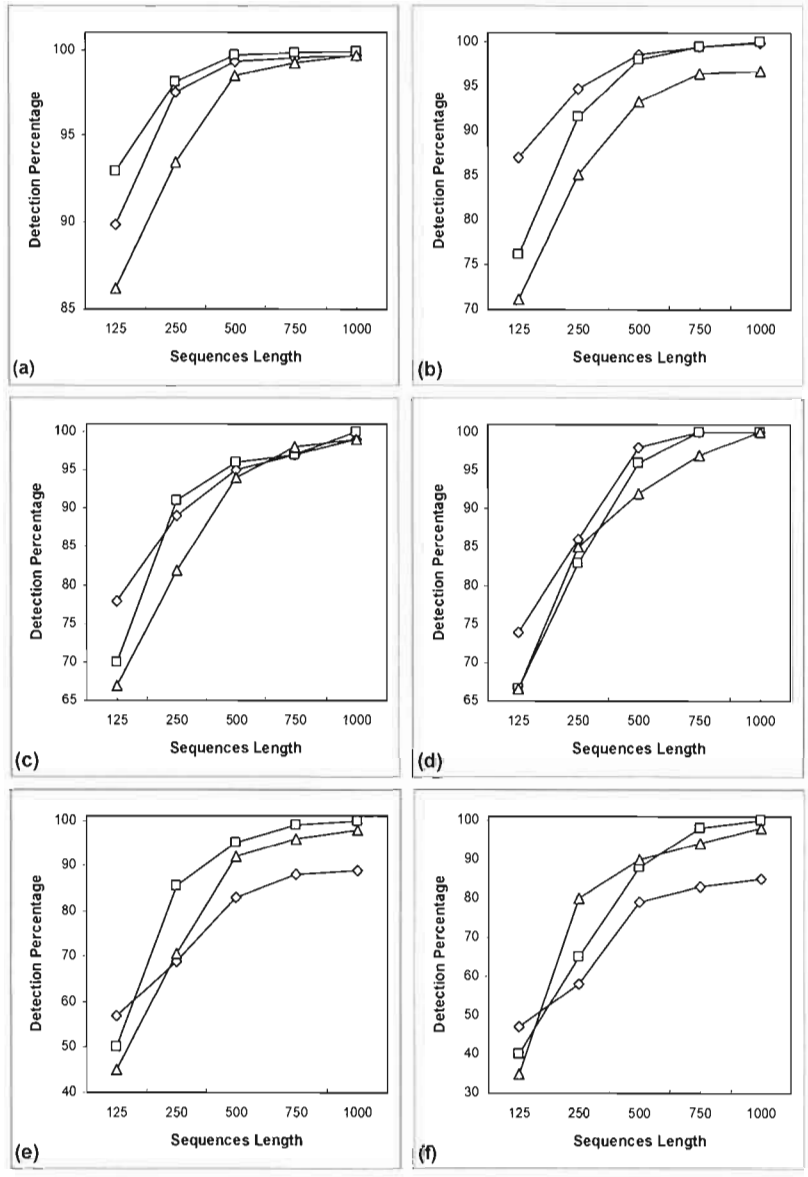


Figure 7. HGT detection rates obtained for random phylogenies with 8 to 64 leaves (8-a, 16-b, 24-b, 32-d, 48-e, 64-f) using the LS function for optimization. Jukes and Cantor (◊), Kimura 2-parameter (□), and Jin-Nei Gamma (Δ) models were used for the tree generation.

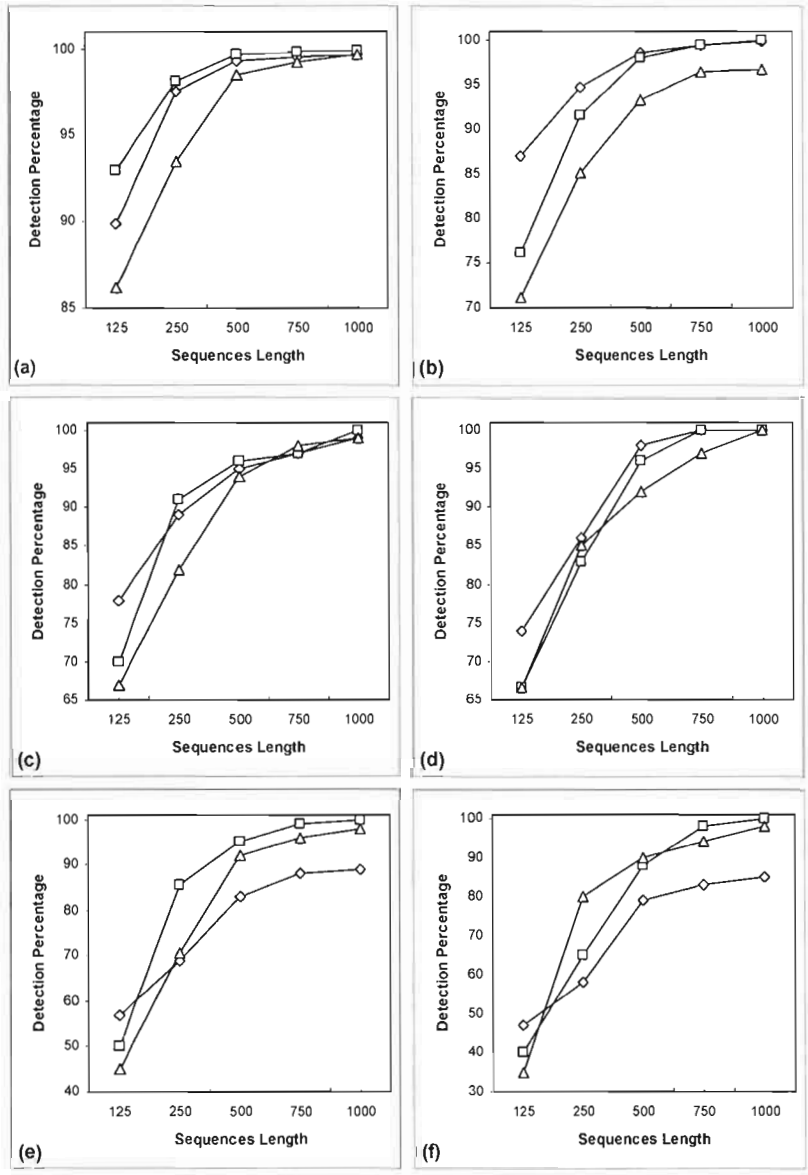


Figure 7. HGT detection rates obtained for random phylogenies with 8 to 64 leaves (8-a, 16-b, 24-b, 32-d, 48-e, 64-f) using the LS function for optimization. Jukes and Cantor (◊), Kimura 2-parameter (□), and Jin-Nei Gamma (Δ) models were used for the tree generation.

Figures 6 and 7 present the average simulation results obtained for random phylogenies with 8 to 64 leaves, using as optimization criteria the RF topological distance and LS function, respectively. These figures illustrate how the detection rate changes as the number of sites varies from 125 to 1000. As expected, the detection rate grows as the number of sites increases and the number of species decreases. Note that for the phylogenies with 8 to 32 leaves the best results were obtained under the Kumura and Jukes-Cantor models. For the phylogenies with 48 to 64 species the best performances were regularly obtained under the Kimura model, whereas the results found under the Jukes-Cantor model were the worst of the three evolutionary models.

This trend can be observed in the case of both optimization criteria. Obviously, with the short sequences we have a bigger phylogenetic error that can either appear like a HGT, when it does not occur, or disguise a real HGT. Tables 1 and 2 (see Appendix) report the false positive and false negative (indicated in parentheses) detection rates obtained using as optimization criteria the RF distance and LS function, respectively. A false positive HGT is an incorrect transfer found by the algorithm and a false negative HGT is the right transfer that has not been detected. A false positive HGT will always occur if the gene tree inferred by NJ (see Step 4 above) is different from the true gene tree (see Step 3 above), but it can also take place when both trees are identical but a transfer going to the direction opposite to the correct HGT disguises it, leading to the same gene tree (see Maddison 1997).

False negative HGTs are mostly due to the error of inferring the gene tree, but can also happen when a transfer going to the opposite direction disguises the correct HGT. As defined, the false positive detection rate is always bigger or equal to the negative one. The analysis of Tables 1 and 2 shows that the false negative rate is almost as big as the false positive rate when the tests were conducted with large phylogenies (48 and 64 species) and short sequences (125 and 250 sites). The false negative rate was noticeably lower than the false positive one in the case of the large phylogenies and long sequences. Furthermore, we have measured the recovery rates for the HGT source, destination, and source and destination combined (i.e. the latter parameter corresponds to the detection rate depicted in Figures 6 and 7). These tests were carried out under the Jukes and Cantor model of sequence evolution and using the RF distance for the algorithmic optimization. Note that the transfer destinations were generally better detectable than their sources. The difference in the source-destination detection was more important for the short sequence. For example, for the sequences with 125 sites it varied, on average, from 6% (for 8 species) to 1% (for 64 species). However, for the longer sequences the source and destination rates were very similar.

Generally, the procedure based on the RF distance provided better results than that based on the LS function. Nevertheless, some noticeable exceptions (e.g. under the Kimura model for the phylogenies with 8 leaves or under the Jin-Nei model in the case of the short sequences) can be pointed out. The simulation study suggested that the accuracy of the transfer detection is highly dependable on the model of sequence evolution, number of considered species, and length of observed sequences.

RESULTS and DISCUSSION

Detecting horizontal transfers of the gene *rpl12e*

We first tested our algorithm on the phylogeny of 14 species of Archaea originally considered by Matte-Tailliez et al. (2002). The latter authors discuss problems encountered when reconstructing some parts of the archaeal phylogeny, pointing out the evidence of HGT events perturbing the evolution of a number of considered genes. Matte-Tailliez et al. inferred the maximum likelihood tree (Figure 9, undirected lines) based on the concatenated 53 ribosomal proteins (7,175 positions) and compared it to the maximum likelihood phylogeny of the gene *rpl12e* (Figure 8) built for the same 14 organisms. The calculations of the best ML tree and its branch lengths for the 53 concatenated proteins were conducted using the PUZZLE program with Γ-law correction.

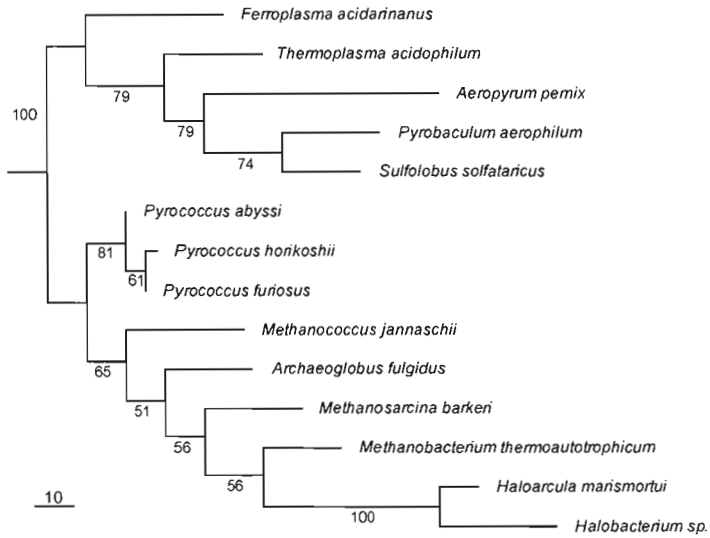


Figure 8. Maximum likelihood phylogenetic tree for the protein *rpl12e* (89 positions). Numbers close to branches are ML bootstrap scores obtained from the sampled protein sequences using the SeqBoot and Proml (JTT model) programs from the PHYLIP package (Felsenstein, 1989). Its topology is identical to the tree found by Matte-Taillez et al. (2002, Figure 3).

Given the topological incongruence of the obtained phylogenies, the authors hypothesized a few cases of lateral transfers of the gene *rpl12e*. More precisely, the case of the transfer between the clades of Thermoplasmatales (*Ferroplasma acidarmanus* and *Thermoplasma acidophilum*) and Crenarchaeota (*Aeropyrum pernix*, *Pyrobaculum aerophilum* and *Sulfolobus solfataricus*) was indicated as the most evident one.

In order to apply our method, we first reconstructed from the original sequences the topologies of the gene (Figure 8) and species trees (Figure 9, undirected lines). The computations were conducted in the framework of the complete gene transfer model, using the RF optimization and subtree constraint options (Figure 1). Five directed branches needed to reconcile the species and gene topologies have been found (Figure

9). The connection representing the transfer between the cluster of *Halobacterium sp.* and *Haloarcula marismortui* and the species *Methanobacterium thermoautotrophicum* was found in the first iteration. This transfer provided the biggest drop of the RF distance between the species and gene phylogenies; its bootstrap score is 55%.

In the second and third iterations, we found the reconciliation branches between the species *Pyrococcus horikoshii* and *Pyrococcus furiosus* and between *Sulfolobus solfataricus* and *Pyrobaculum aerophilum*. Both of these reconciliation branches link closely related species. Such kind of connections may be due to HGT as well as to local topological rearrangements necessary because of the tree reconstruction artifacts (e.g. attraction of long branches, unequal evolutionary rates, etc). The transfer branches 4 and 5 linking the cluster of Crenarchaeota to the species *Thermoplasma acidophilum* and *Ferroplasma acidarmanus* can be interpreted as HGT events that might have taken place between Thermoplasmatales and Crenarchaeota.

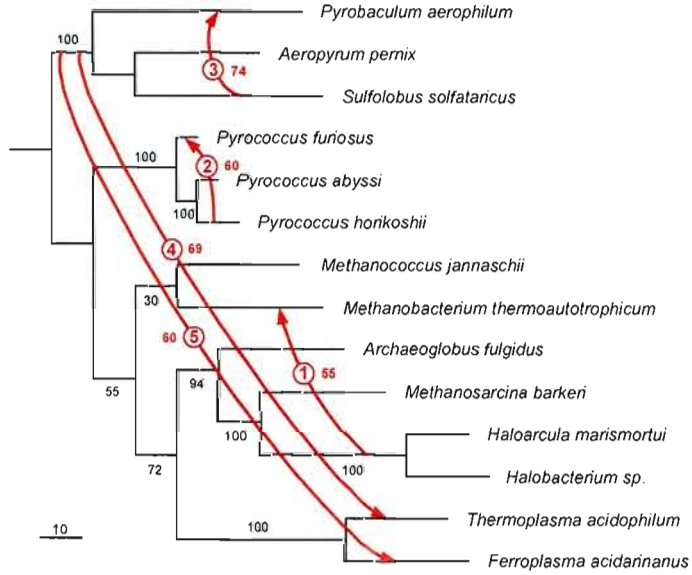


Figure 9. Species tree (Matte-Taillez et al. 2002, Figure 1a) with five reconciliation branches (denoted by arrows). Numbers close to branches are ML bootstrap scores computed by the RELL method upon 2,000 top-ranking trees using the MOLPHY program without correction for among-site variation. Numbers on HGT arrows indicate their order of appearance in the unique gene transfer scenario found by the HGT detection method. Bootstrap scores for transfers are indicated by numbers close to arrow circles. Arrows 4 and 5 depict the HGTs between the clades of Thermoplasmatales and Crenarchaeota also predicted by Matte-Taillez et al. (2002).

In the second and third iterations, we found the reconciliation branches between the species *Pyrococcus horikoshii* and *Pyrococcus furiosus* and between *Sulfolobus solfataricus* and *Pyrobaculum aerophilum*. Both of these reconciliation branches link closely related species. Such kind of connections may be due to HGT as well as to local topological rearrangements necessary because of the tree reconstruction artifacts (e.g. attraction of long branches, unequal evolutionary rates, etc). The transfer branches 4 and 5 linking the cluster of Crenarchaeota to the species *Thermoplasma acidophilum* and *Fer-*

roplasma acidarmanus can be interpreted as HGT events that might have taken place between Thermoplasmatales and Crenarchaeota.

Note, that HGT between these two groups was also predicted by Matte-Taillez et al. (2002). In fact, the transfers 4 and 5 could consist of a unique transfer between the clades of Thermoplasmatales and Crenarchaeota that was separated into two transfers by our method due to the application of the subtree constraint (Figure 1) and the presence of the tree reconstruction artifacts. Figure 10 illustrates the evolution of the newly formed Thermoplasmatales-Crenarchaeota clade involving the HGTs 4 and 5. The usage of the LS criterion instead of RF leads to the solution consisting of 6 HGTs including all transfers from Figure 9 except the HGT number 2 that goes in the opposite direction. Note that a new reconciliation branch found with LS brings the species *Methanococcus jannaschii* to the cluster of 4 species including *Archaeoglobus fulgidus*. This reconciliation branch turns out to be useless and have a low bootstrap score of 14%.

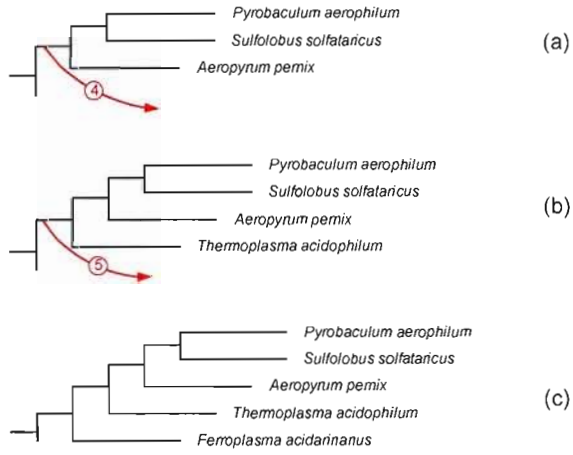


Figure 10. Changes in the Crenarchaeota-Thermoplasmatales cluster occurring after the addition of HGT branches 4 and 5. (a) This cluster after the transfer 3; the species *Thermoplasma acidophilum* joins the Crenarchaeota cluster. (b) This cluster after the transfer 4; the species *Ferroplasma acidarmanus* is added to the clade comprising three Crenarchaeota and *Thermoplasma acidophilum*. (c) This cluster after the transfer 5.

CONCLUSION

We presented two polynomial-time algorithms for detecting horizontal gene transfer events. We considered the complete and partial gene transfer models, implying at each step, either the transformation of a species phylogeny into another tree or its transformation into a network structure. The algorithm for inferring complete gene transfers exploits the discrepancies between the species and gene phylogenies either to map the gene tree into the species tree by least-squares or to compute a topological distance between them and then estimate the possibility of a HGT event between each pair of

branches of the species phylogeny. The models based on the optimization of the least-squares function and the Robinson and Foulds topological distance were introduced.

Inferred HGTs should be carefully analyzed using all available information about the data in hand in order to select the transfers that will be represented as a final solution. Each gene transfer branch added to the species phylogeny aids to resolve a conflict between it and the gene tree (i.e. helps to reconcile the species and gene phylogenies). A bootstrap validation procedure allowing one to assess the reliability of a specific gene transfer or whole gene transfer scenario was proposed. A comprehensive Monte Carlo study was carried out to test the ability of the new method to recover correct HGTs. It provided very encouraging results especially when the Robinson and Foulds distance was used as an optimization criterion. The example of the evolution of the gene *rpl12e* was considered in the application section. More simulation work is required to investigate the properties of the algorithm intended to infer partial gene transfers.

As any method of phylogenetic inferring, the new HGT detection method is subject to a number of artifacts which generally affect phylogenetic analysis; the main of them being: Attraction of long branches, unequal evolutionary rates, and situations when the occurrence of some HGT events almost coincides with speciation events located closely to the recipient species. It is important to investigate in greater details the impact of these artifacts on the HGT detection technique introduced in this article. It would be also interesting to extend the presented model to the case, where the gene and species trees have different numbers of taxa; this situation can take place when some species have more than one copy of the gene under consideration.

The software implementing the new algorithms for detecting complete and partial horizontal gene transfers is freely available at the following URL address: < http://www.info2.uqam.ca/~boca05/software/> (this is a consol version running on the Unix and Windows platforms; it is distributed along with its C++ source code). A graphical version of this program has been also implemented and included in the T-Rex web server (Makarenkov 2001) at the following URL: < http://www.trex.uqam.ca >.

ACKNOWLEDGEMENTS

The authors are grateful to Dr. Hervé Philippe for his help in the analysis of the *rpl12e* data.

REFERENCES

Barthélemy, J.-P. and Guénoche, A.: *Les arbres et les représentations des proximités*, Paris: Masson (1988)

Barthelemy J.-P. and Guenoche A.: *Trees and proximity representations.* New York: Wiley (1991)

Boc, A., Makarenkov, V.: New Efficient Algorithm for Detection of Horizontal Gene Transfer Events. In: Benson, G. and Page, R. (eds.): *Algorithms in Bioinformatics.* WABI. LNCS. Springer Verlag, Budapest (2003) 190-201

Charleston, M. A.: Jungle: a new solution to the host/parasite phylogeny reconciliation problem. *Math. Bioscience* (1998) 149:191-223

Denamur, E., Lecointre, G. and Darlu, P. et al. (12 co-authors): Evolutionary implications of the frequent horizontal transfer of mismatch repair genes. *Cell* (2000) 103:711-721

Doolittle, W. F.: Phylogenetic classification and the universal tree. *Science* (1999) 284:2124-2129

Felsenstein, J.: Confidence limits on phylogenies: An approach using the bootstrap. *Evolution* (1985) 39:738-791

Felsenstein, J. PHYLIP - Phylogeny Inference Package (Version 3.2) *Cladistics* (1989) 5:164-166

Guindon, S. and Gascuel, O.: Efficient biased estimation of evolutionary distances when substitution rates vary across sites. *Mol. Biol. Evol.*, (2002) 19:534-543

Hallett, M. and Lagergren, J.: Efficient algorithms for lateral gene transfer problems. In: El-Mabrouk,N., Lengauer,T., Sankoff,D. (eds.): *RECOMB*, ACM, New-York (2001) 149-156

Hallett, M., Lagergren, J., Tofigh, A.: Simultaneous identification of duplications and lateral transfers. In: Bourne, P.E. and Gusfield, D. (eds.): *RECOMB*, ACM, San Diego (2004) 347-356

Hein, J., Jiang, T., Wang, L., Zhang, K.: On the complexity of comparing evolutionary trees. *Discr. Appl. Math.* (1996) 71:153-169

Jin, L. and Nei, M.: Limitations of the evolutionary parsimony method of phylogenetic analysis *Mol. Biol. Evol.*, (1990) 7: 82-102

Jukes, T.H. and Cantor, C.: Mammalian Protein Metabolism. In *Evolution of protein molecules* (H. N. Munro, editor). New York: Academic Press. . (1969) 21-132

Kimura, M. A: Simple method for estimating evolutionary rate of base substitutions through comparative studies of nucleotide sequences. *J. Mol. Evol.* (1980) 16:111-120

Kuhner, M., Felsenstein, J.: A simulation comparison of phylogeny algorithms under equal and unequal evolutionary rates. *Mol. Biol. Evol.* (1994) 11:459-468

Lawrence, J. G., Ochman, H.: Amelioration of bacterial genomes: rates of change and exchange. *J. Mol. Evol.* (1997) 44:383-397

Legendre, P. (Guest Editor): Special section on reticulate evolution. *J. of Classif* (2000) 17: 153-195

Legendre, P. and Makarenkov, V.: Reconstruction of biogeographic and evolutionary networks using reticulograms. *Syst. Biol.* (2002) 51:199-216

Maddison, W. P.: Gene trees in species trees. *Syst. Biol.* (1997) 46:523-536

Makarenkov, V. and Leclerc, B.: An algorithm for the fitting of a phylogenetic tree according to a weighted least-squares criterion. *J. of Classif* (1999) 16:3-26

Makarenkov, V. T-Rex: reconstructing and visualizing phylogenetic trees and reticulation networks. *Bioinformatics* (2001) 17:664–668

Makarenkov, V., Boc, A., Delwiche, C. F., Diallo, A.B. and Philippe, H.: New efficient algorithm for modeling partial and complete gene transfer scenarios. Data Science and Classification, V. Batagelj, H.-H. Bock, A. Ferligoj, and A. Ziberna (Eds.), IFCS 2006, Series: *Studies in Classification, Data Analysis, and Knowledge Organization*, Springer Verlag, (2006) 341-349

Matte-Tailliez, O., Brochier, C., Forterre, P. and Philippe, H.: Archaeal phylogeny based on ribosomal proteins. *Mol. Biol. Evol.* (2002) 19:631-639

Mirkin, B., Fenner T. I., Galperin M. and Koonin E.: Algorithms for computing parsimonious evolutionary scenarios for genome evolution, the last universal common ancestor and dominance of horizontal gene transfer in the evolution of prokaryotes. *BMC Evol. Biol.* 3:2 (2003)

Mirkin, B.: Mapping gene family data onto evolutionary trees, in M. Chavent, O. Dordan, C. Lacomblez, M. Langlais, and B. Patouille (Eds.), *Proceedings of the 11th Rencontres de la Société Francophone de Classification*, University of Bordeaux, (2004) 61-68

Moret, B. M. E., Nakhleh, L., Warnow, T., Linder, C. R., Tholse, A., Padolina, A., Sun, J. and Timme R.: Phylogenetic networks: modeling, reconstructibility, and accuracy. IEEE/ACM Trans. on Comput. *Biol. and Bioinf.* (2004) 1:13-23

Nakhleh, L., Ruths, D. and Innan, H.: Gene trees, species trees, and species networks. In: Guerra, R. and Allison, D. (eds.): *Meta-analysis and combining information in genetics* (2005) 1-27

Page, R.D.M.: Maps between trees and cladistic analysis of historical associations among genes, organisms, and areas. *Systematic Biology* (1994) 43:58-77.

Page, R.D.M. and Charleston, M.A.: From gene to organismal phylogeny: Reconciled trees *Bioinformatics* (1998a) 14:819-820

Page, R. D. M. and Charleston, M. A.: Trees within trees: phylogeny and historical associations. *Trends Ecol. Evol.* (1998b) 13:356-359

Rambaut, A. and Grassly, N.C.: Seq-Gen: An application for the Monte Carlo simulation of DNA sequence evolution along phylogenetic trees. *Comput. Appl. Biosci.* (1996)

Robinson, D. R. and Foulds, L. R.: Comparison of phylogenetic trees. *Math. Biosciences* (1981) 53:131-147

Saitou, N. and Nei, M.: The neighbor-joining method: a new method for reconstructing phylogenetic trees. *Mol. Biol. Evol.* (1987) 4:406-425

von Haeseler, A., Churchill, G. A.: Network models for sequence evolution. *J. Mol. Evol.* (1993) 37:77-85

APPENDIX

This Appendix includes the results of the tests described in the section Simulation Study. The results reported in Tables 1 and 2 correspond to the graphics represented in Figures 6 (optimization using the RF distance) and 7 (optimization using the LS function). They were obtained from simulations carried out for random binary phylogenies with 8, 16, 24, 32, 48, and 64 leaves, whereas the sequence length varied from 125 to 1000 sites. Note that the sum of the HGT detection rate shown in Figures 6 and 7 and of the false negative detection rate reported in Tables 1 and 2 is always 100%.

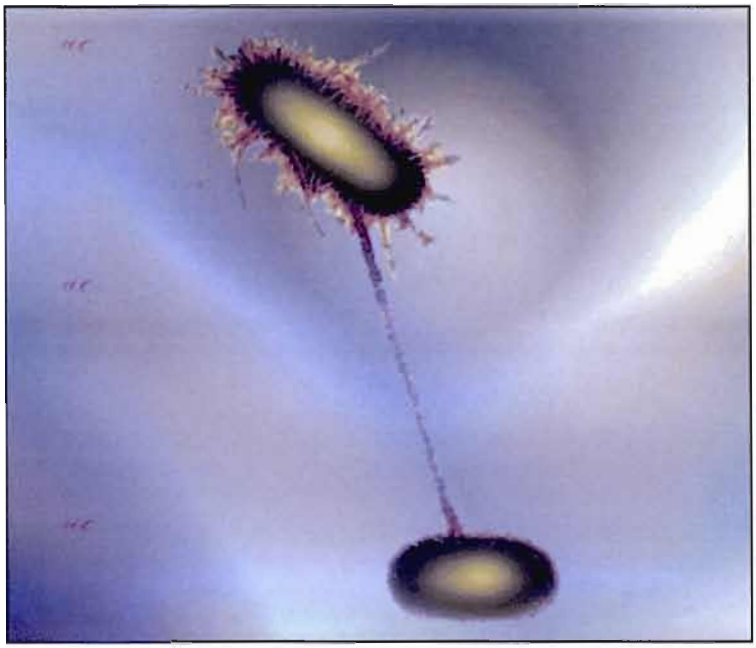
Table 1. False positive and false negative (in parentheses) detection rates obtained for random phylogenies with 8 to 64 leaves using the RF distance as an optimization criterion. A false positive HGT is an incorrect transfer found by the algorithm and a false negative HGT is the right transfer that has not been found. For each sequence length, the simulations were carried out for 500 random phylogenies with 8 and 16 leaves and 100 random phylogenies with 24 to 64 leaves.

RF rates (in %)			Sequence length				
			125	250	500	750	1000
Species number	8	Jukes-Cantor	14.9 (7.8)	5.9 (3.5)	1.1 (0.7)	0.3 (0.3)	0.0 (0.0)
		Kimura	12.9 (8.7)	3.3 (2.2)	0.2 (0.1)	0.1 (0.1)	0.0 (0.0)
		Jin-Nei	20.1 (15.0)	3.9 (2.5)	1.6 (1.3)	1.1 (1.1)	0.5 (0.5)
	16	Jukes-Cantor	25.7 (14.0)	7.1 (4.5)	1.2 (0.7)	0.4 (0.3)	0.0 (0.0)
		Kimura	35.1 (22.5)	11.9 (7.9)	3.2 (2.3)	0.6 (0.6)	0.1 (0.1)
		Jin-Nei	43.0 (30.0)	22.5 (16.5)	7.6 (6.6)	5.3 (4.9)	2.3 (2.3)
	24	Jukes-Cantor	36 (18)	15 (10)	4 (3)	1 (1)	1 (1)
		Kimura	43 (24)	24 (13)	4 (2)	2 (0)	0 (0)
		Jin-Nei	55 (35)	33 (18)	19 (10)	9 (6)	5 (4)
	32	Jukes-Cantor	37 (20)	29 (11)	4 (2)	1 (1)	1 (0)
		Kimura	60 (35)	31 (14)	8 (3)	3 (1)	2 (0)
		Jin-Nei	70 (38)	47 (25)	16 (9)	8 (3)	8 (3)
	48	Jukes-Cantor	65 (48)	49 (29)	28 (15)	27 (9)	25 (8)
		Kimura	55 (38)	46 (18)	9 (3)	3 (1)	1 (1)
		Jin-Nei	70 (40)	58 (24)	19 (8)	9 (2)	4 (1)
	64	Jukes-Cantor	70 (60)	45 (35)	27 (17)	23 (13)	20 (10)
		Kimura	65 (55)	35 (25)	14 (4)	12 (2)	10 (0)
		Jin-Nei	60 (50)	44 (34)	22 (12)	18 (8)	14 (4)

Table 2. False positive and false negative (in parentheses) detection rates obtained for random phylogenies with 8 to 64 leaves using the LS function as an optimization criterion. A false positive HGT is an incorrect transfer found by the algorithm and a false negative HGT is the right transfer that has not been found. For each sequence length, the simulations were carried out for 500 random phylogenies with 8 and 16 leaves and 100 random phylogenies with 24 to 64 leaves.

LS rates (in %)			Sequence length				
			125	250	500	750	1000
Species number	8	Jukes-Cantor	17.2 (10.1)	5.0 (2.5)	0.8 (0.7)	0.8 (0.5)	0.3 (0.3)
		Kimura	10.8 (7.0)	2.8 (1.9)	0.3 (0.3)	0.2 (0.2)	0.1 (0.1)
		Jin-Nei	18.6 (13.8)	7.8 (6.5)	1.7 (1.5)	0.9 (0.8)	0.5 (0.3)
	16	Jukes-Cantor	25.5 (13.0)	7.6 (5.3)	2.2 (1.4)	0.8 (0.5)	0.1 (0.1)
		Kimura	37.6 (23.8)	11.9 (8.4)	2.3 (2.0)	0.6 (0.6)	0.0 (0.0)
		Jin-Nei	40.9 (28.8)	20.9 (14.8)	8.1 (6.7)	3.8 (3.6)	3.3 (3.3)
	24	Jukes-Cantor	43 (22)	13 (11)	5 (5)	3 (3)	1 (1)
		Kimura	59 (30)	26 (9)	7 (4)	4 (3)	1 (0)
		Jin-Nei	67 (33)	26 (18)	12 (6)	6 (2)	3 (1)
	32	Jukes-Cantor	47 (26)	21 (14)	5 (2)	0 (0)	0 (0)
		Kimura	56 (33)	31 (17)	9 (4)	0 (0)	0 (0)
		Jin-Nei	50 (33)	31 (15)	12 (8)	11 (3)	4 (0)
	48	Jukes-Cantor	53 (43)	38 (31)	33 (17)	22 (12)	19 (11)
		Kimura	60 (50)	34 (14)	16 (5)	5 (1)	2 (0)
		Jin-Nei	65 (55)	50 (29)	25 (8)	12 (4)	10 (2)
	64	Jukes-Cantor	63 (53)	52 (42)	41 (21)	27 (17)	25 (15)
		Kimura	70 (60)	45 (35)	22 (12)	15 (2)	10 (0)
		Jin-Nei	75 (65)	40 (20)	20 (10)	16 (6)	12 (2)

Cover image of Systematic Biology (March 2010)



Cover Illustration: Horizontal gene transfer (HGT) is one of the main mechanisms contributing to microbial genome diversification (see the article by Boc, Philippe and Makarenkov in this issue). It is rampant among various groups of genes in bacteria. HGT poses several risks to humans, including: antibiotic-resistant genes spreading to pathogenic bacteria, transgenic DNA inserting into human cell and triggering cancer, and disease-associated genes spreading and recombining to create new viruses and bacteria. Bacteria and Archaea have sophisticated mechanisms for the acquisition of new genes through HGT, which may have been favored by natural selection as a more rapid mode of adaptation than the alteration of gene functions through numerous point mutations. The three main types of HGT are the following: transformation, consisting of uptake of naked DNA from the environment, conjugation that is mediated by conjugal plasmids or transposons, and transduction, consisting of DNA transfer by phage. A bacterial conjugation plasmid transfer is shown. Photo credit AJC1 Flickr.

Syst. Biol. 59(2):195–211, 2010

DOI:10.1093/sysbio/syp103
Advance Access publication on January 21, 2010

Inferring and Validating Horizontal Gene Transfer Events Using Bipartition Dissimilarity

ALIX BOC[1], HERVÉ PHILIPPE[2], AND VLADIMIR MAKARENKOV[1,*]

[1]*Département d'informatique, Université du Québec à Montréal, C.P. 8888, Succ. Centre-ville, Montréal, Québec, Canada H3C 3P8; E-mail: boc.alix@courrier.uqam.ca; and*
[2]*Département de biochimie, Faculté de Médecine, Université de Montréal, C.P. 6128, Succ. Centre-ville, Montréal, Québec, Canada H3C 3J7; E-mail: herve.philippe@umontreal.ca;*
**Correspondence to be sent to: Département d'informatique, Université du Québec à Montréal, C.P. 8888, Succ. Centre-Ville, Montréal (Québec), Canada, H3C 3P8; E-mail: makarenkov.vladimir@uqam.ca.*

Received 23 October 2008; reviews returned 25 March 2009; accepted 12 November 2009
Associate Editor: Olivier Gascuel

Abstract.—Horizontal gene transfer (HGT) is one of the main mechanisms driving the evolution of microorganisms. Its accurate identification is one of the major challenges posed by reticulate evolution. In this article, we describe a new polynomial-time algorithm for inferring HGT events and compare 3 existing and 1 new tree comparison indices in the context of HGT identification. The proposed algorithm can rely on different optimization criteria, including least squares (LS), Robinson and Foulds (RF) distance, quartet distance (QD), and bipartition dissimilarity (BD), when searching for an optimal scenario of subtree prune and regraft (SPR) moves needed to transform the given species tree into the given gene tree. As the simulation results suggest, the algorithmic strategy based on BD, introduced in this article, generally provides better results than those based on LS, RF, and QD. The BD-based algorithm also proved to be more accurate and faster than a well-known polynomial time heuristic RIATA-HGT. Moreover, the HGT recovery results yielded by BD were generally equivalent to those provided by the exponential-time algorithm LatTrans, but a clear gain in running time was obtained using the new algorithm. Finally, a statistical framework for assessing the reliability of obtained HGTs by bootstrap analysis is also presented. [Bipartition dissimilarity; bootstrap analysis; horizontal gene transfer; least squares; phylogenetic tree; quartet distance; Robinson and Foulds topological distance.]

The understanding that horizontal gene transfer (HGT, also called lateral gene transfer) might have played a key role in species evolution is one of the most fundamental changes in our perception of general aspects of molecular biology (Doolittle et al. 2003; Koonin 2003). HGT is a direct transfer of genetic material from one lineage to another. Bacteria and archaea have sophisticated mechanisms for the acquisition of new genes through HGT, which may have been favored by natural selection as a more rapid way of adaptation than the alteration of gene functions through numerous point mutations (Doolittle 1999; Gogarten et al. 2002; Zhaxybayeva et al. 2004). The 3 main types of HGT are the following: transformation, consisting of uptake of naked DNA from the environment, conjugation that is mediated by conjugal plasmids or transposons, and transduction, consisting of DNA transfer by phage.

There are 2 main approaches to identify the genes that have been transferred horizontally. First, sequence analysis of the host genome may reveal areas with GC content or codon usage patterns atypical for it (Lawrence and Ochman 1997). Assuming that these sequences have not arisen from a selective process means that they might have been acquired horizontally. Tsirigos and Rigoutsos (2005) discussed a method for detecting HGT that relies on a gene's nucleotide composition and obviates the need for knowledge of codon boundaries. Second, the comparison of a morphology-based species tree or molecular tree based on a molecule that is assumed to be refractory to HGT (e.g., 16S rRNA or 23S rRNA) against a phylogeny of an observed gene, inferred for the same set of organisms, may reveal topological conflicts that can be explained by HGT. Ribosomal genes may also undergo HGT, but they seem to do it at a relatively low rate, and can serve as a first approximation to a species (i.e., organismal) phylogeny in the absence of other data (Acinas et al. 2004).

The latter approach includes numerous methods that started to appear in the early 1990s. First methods using network-based models to recover HGT were proposed by Hein (1990), von Haeseler and Churchill (1993), Page (1994), and Page and Charleston (1998). Mirkin et al. (1995) put forward a tree reconciliation method that combines different gene trees into a unique organismal phylogeny. The paper by Moret et al. (2004) presents an overview of the network modeling in phylogenetics. Maddison (1997) and Page and Charleston (1998) first described the set of evolutionary rules that should be taken into account when modeling HGT.

Several recently proposed methods deal with approximation of the subtree prune and regraft (SPR) distance that is closely related to the inference of HGT events. Bordewich and Semple (2004) showed that computing the SPR distance between rooted binary trees is NP-hard. A HGT model allowing for mapping numerous gene trees into a species tree was described by Hallett and Lagergren (2001; LatTrans algorithm). The LatTrans algorithm generates all shortest SPR scenarios but is exponential in the number of transfers. On the other hand, Boc and Makarenkov (2003) proposed an algorithm that can be suitable for inferring partial HGTs. Mirkin et al. (2003) designed an algorithm for the reconciliation of phyletic patterns with a species tree by postulating gene loss, gene emergence, and HGT. The latter authors showed that in each situation, their algorithm provides a parsimonious evolutionary scenario consisting of mapping gene loss and gain events into a species phylogenetic tree. Hallett et al. (2004) introduced

a combinatorial model incorporating HGT and duplication events. The "HorizStory" algorithm intended to approximate the SPR distance between rooted and possibly nonbinary phylogenetic trees was described by MacLeod et al. (2005). The algorithm works by, first, eliminating identical rooted subtrees in the gene and species trees. SPR moves are then carried out recursively on the remaining trees until they are brought into agreement. Beiko and Hamilton (2006) described the efficient evaluation of edit paths (EEEP) algorithm searching for a minimum number of SPR operations between 2 rooted trees. The approach adopted by EEEP considers the bipartitions induced by the branches of the reference and test trees. The key to topological comparisons in this algorithm is the subdivision of the reference tree bipartitions into those that are concordant and discordant with respect to the test tree. On the other hand, Nakhleh et al. (2005) developed the "RIATA-HGT" heuristic based on the divide-and-conquer approach. Than and Nakhleh (2008) showed that the latest version of RIATA-HGT is considerably faster than LatTrans while being almost equivalent in terms of accuracy. Recently, first probabilistic and parsimony models of HGT have started to appear. Thus, Csürös and Miklós (2006) introduced a Markov model of evolution of a gene family along a phylogenetic tree. It includes parameters for the rates of HGT, gene duplication, and gene loss. Jin et al. (2006, 2007) described 2 new algorithms for inferring HGT events in the framework of the maximum likelihood (ML) and maximum parsimony models.

In this article, we describe a new accurate algorithm for inferring and validating HGT events. First, we will introduce and study the "bipartition dissimilarity" (BD) between 2 phylogenies. This measure of proximity between 2 phylogenetic trees can be considered as a refinement of the Robinson and Foulds (RF) distance (Robinson and Foulds 1981), which takes into account only identical bipartitions in the compared phylogenies. We will show that the use of the BD as an optimization criterion offers important improvements over the well-known least squares (LS), RF, and quartet distance (QD) measures. A bootstrap validation procedure for assessing the reliability of obtained HGTs will be also presented. Then, a comparison of the performances of the BD-based algorithm with LatTrans (Hallett and Lagergren 2001) and RIATA-HGT (Nakhleh et al. 2005; Than and Nakhleh 2008) will be made in terms of both HGT recovery and running time and followed by 2 application examples.

Materials and Methods

BD and Other Optimization Criteria

The new algorithm for identifying HGTs proceeds by a progressive reconciliation of the given rooted species and gene phylogenies denoted by T and T', respectively. At each step of the algorithm, several pairs of branches in T are tested against the hypothesis that a HGT has occurred between them. The considered HGT model works in both cases: 1) when the transferred gene supplants the orthologous gene of the recipient genome and 2) when the transferred gene, absent in the recipient genome, is added to it. Thus, the original species phylogenetic tree T is gradually transformed into the gene phylogenetic tree T' by a series of SPR moves (i.e., HGTs). The goal is to find the shortest sequence of trees $T, T_1, T_2, \ldots, T'$ that transforms T into T'. A number of necessary "evolutionary constraints" should be taken into account because postulating a HGT requires that the source and destination species are contemporaneous. For instance, the transfers within the same lineage (Fig. A1a in Appendix 1) and the transfers that are crossing as shown in Figure A1b–d are leading to inappropriate HGT scenarios and must be prohibited (see also Maddison 1997; Page and Charleston 1998; or Hallett and Lagergren 2001).

The problem of calculating the SPR distance is known to be NP-hard for both rooted and unrooted trees. The first proof of NP-hardness, in the case of unrooted trees, was given by Hein et al. (1996), but was found to be incorrect by Allen and Steel (2001), who showed that the related tree bisection and reconnection distance problem is NP-hard but fixed parameter tractable for unrooted binary trees. Then, Hickey et al. (2008) provided a correct complete proof of the NP-hardness in the case of unrooted trees. On the other hand, Bordewich and Semple (2004) proved the NP-hardness of the computation of the SPR distance for rooted binary trees.

We consider 4 optimization criteria that can be used to select the best HGT at each algorithmic step. The first of them is the LS function. It is computed as follows:

$$\mathrm{LS} = \sum_i \sum_j (d(i,j) - \delta(i,j))^2, \qquad (1)$$

where $d(i,j)$ is the patristic distance between the leaves (i.e., species or taxa) i and j in the species tree T (or in the transformed species tree T_k obtained from T after the SPR operation number k) and $\delta(i,j)$ is the patristic distance between i and j in the gene tree T'. The second criterion that can be used for assessing the discrepancy between the species and gene phylogenies is the RF topological distance (Robinson and Foulds 1981). This distance equals to the minimum number of elementary operations, consisting of merging and splitting nodes necessary to transform one tree into the other. The third considered criterion, the QD, is the number of quartets, subtrees induced by 4 leaves, that differs between the compared trees. We can use these criteria as follows to determine the best HGT. When several transformations of the species tree, consisting of SPR moves between its subtrees, are evaluated, the SPR move providing the minimum of the selected criterion computed for the transformed species tree T_1 and the gene tree T' is retained.

The fourth optimization criterion is the BD defined as follows. Without loss of generality, we assume that T and T' are binary phylogenetic trees having the same set of leaves. A bipartition vector (i.e., split or bipartition) of the tree T is a binary vector induced by a branch of T. Let

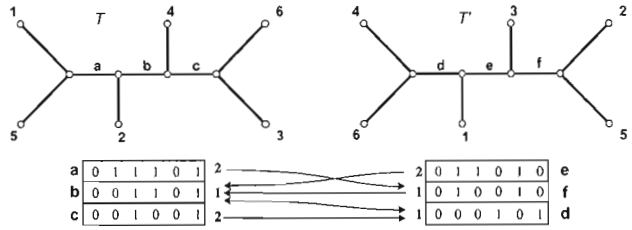


FIGURE 1. Trees T and T' and their bipartition tables. Each line of the bipartition table corresponds to an internal branch of the tree. Arrows indicate the associations between the bipartition vectors in the 2 tables. Value in bold close to each vector represents the associated distance.

BT be the bipartition table of the "internal branches" of the tree T (i.e., the table including all bipartition vectors induced by internal branches of T) and BT′ be the bipartition table of the internal branches of the tree T'. The bipartition dissimilarity bd between T and T' is computed as follows:

$$bd = \left(\sum_{a \in \mathrm{BT}} \underset{b \in \mathrm{BT}'}{Min}(Min(d(a,b)\,;d(a,\bar{b}))) + \sum_{b \in \mathrm{BT}'} \underset{a \in \mathrm{BT}}{Min}(Min(d(b,a)\,;d(b,\bar{a}))) \right) \Big/ 2, \quad (2)$$

where $d(a,b)$ is the Hamming distance between the bipartition vectors a and b, and $\bar{a}$ and $\bar{b}$ are the complements of a and b, respectively. Such a measure represents a refinement of the RF metric, which takes into account only identical bipartitions.

For instance, the BD between the trees T and T' with 6 leaves (Fig. 1) is computed as follows: $bd(T,T')=((2+1+2)+(2+1+1))/2=4.5$. Here, the minimum of the Hamming distance between the bipartition corresponding to the branch a and all the bipartition vectors in BT′ is 2. It is the distance between the vectors a and $\bar{f}$, and a and d (in Fig. 1, only the association between a and $\bar{f}$ is presented by an arrow). For the bipartition b, this distance is 1 (the distance between b and d) and for the bipartition c, this distance is 2 (the distance between c and d). In the same way, the minimum distance between the bipartition e and all the bipartitions in BT is 2 (with $\bar{b}$), for the bipartition f, this distance is 1 (also with $\bar{b}$), and for the bipartition d, it is 1 (with b).

This example shows that several bipartition vectors of the first bipartition table can be associated with the same bipartition vector of the second table; for example, d, e, and f are associated with b (or $\bar{b}$), and both b and c are associated with d (Fig. 1). Moreover, the BD is not always a metric. For trees with 5 or more leaves, one can exhibit 3 tree topologies for which the triangle inequality does not hold. Propositions 1 and 2 below (their proofs can be found in online Appendix 1 available from http://www.sysbio.oxfordjournals.org/) establish some interesting properties of the BD. Thus, Proposition 1 states the sufficiency condition ensuring that a BD satisfies the triangle inequality (and is a metric), whereas Proposition 2 establishes the maximum values of BD depending on the number of tree leaves. The bipartition a of a tree T is associated with the bipartition b of the tree T' (this association is denoted by $a \rightarrow b$) if the Hamming distance between the bipartition vectors corresponding to a and b is the smallest among all possible distances computed between a and all the bipartition vectors of T'.

Proposition 1 *Let T_1, T_2, and T_3 be phylogenetic trees with the same number of internal branches and the same sets of leaves. Then, if:*

1. *For any 2 bipartitions, a and b from different trees: $a \rightarrow b$ implies that $b \rightarrow a$ and*
2. *For any 3 bipartitions, $a \in T_1, b \in T_2$, and $c \in T_3 : a \rightarrow b$ and $b \rightarrow c$ implies that $a \rightarrow c$,*

then the triangle inequality, $bd(T_1,T_2) \leq bd(T_1,T_3) + bd(T_2,T_3)$, holds.

Proposition 2 *The value of the BD between 2 phylogenetic trees on the same sets of n leaves ranges from 0 to $n(n-3)/2$ if n is even and from 0 to $(n-1)(n-3)/2$ if n is odd.*

Heuristic Algorithm for Predicting HGTs

In this section, we discuss the main features of the new algorithm for inferring HGTs. Consider a HGT in the species tree T going from a to b and transforming it into the tree T_1 (Fig. 2). The following constraint is postulated: To permit the HGT between the branches (x,y)

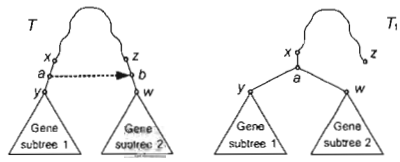


FIGURE 2. Subtree constraint: The transfer between the branches (x,y) and (z,w) in the species tree T is allowed if and only if the cluster rooted by the branch (x,a), and regrouping both affected subtrees, is present in the gene tree. Throughout the article, a single tree branch is depicted by a plane line and a path is depicted by a wavy line.

and (z, w) of the species tree T, the cluster (i.e., clade) consisting of the subtree rooted by the branch (x, a), and including the vertices y and w in the tree T_1, must be present in the gene tree T'.

Such a constraint, called here the "subtree constraint," enables us to arrange first the topological conflicts between T and T', which are due to the transfers between the closest ancestors of the contemporary species, and which are easier to detect, and then identify the HGTs that occurred deeper in the phylogeny. Moreover, the use of the subtree constraint allows us to take into account automatically all required evolutionary constraints (Fig. A1) because both subtrees involved in the HGT have to be present in the gene tree T' as well as the new subtree that they form after the transfer (Fig. 2). Indeed, if the same lineage HGTs (Fig. A1a) or the transfers crossing in a way presented in Figure A1(b–d) were permitted, the subtree constraint would not hold (the reader is referred to the Discussion section where all advantages brought by this constraint are summarized).

The 2 following theorems establish some properties of bipartitions in the context of HGTs satisfying the subtree constraint. These properties are used in the HGT detection algorithm described below. The proofs of both theorems are presented in online Appendix 1.

Theorem 1. *If the newly formed subtree* Sub_{yw} *resulting from the HGT (i.e., the subtree rooted by the branch* (x, a) *in Fig. 2) is present in the gene tree* T'*, and the bipartition vector associated with the branch* (x, x_1) *in the transformed species tree* T_1 *(Fig. A2) is present in the bipartition table of* T'*, then the HGT from* (x, y) *to* (z, w)*, transforming* T *into* T_1*, is a part of a minimum cost HGT scenario transforming* T *into* T' *and satisfying the subtree constraint.*

Theorem 2. *If the newly formed subtree* Sub_{yw} *resulting from the HGT (i.e., the subtree rooted by the branch* (x, a) *in Fig. 2) is present in the gene tree* T'*, and all the bipartition vectors associated with the branches of the path* (x', z') *in the transformed species tree* T_1 *(Fig. A2) are present in the bipartition table of* T'*, and the path* (x', z') *in* T_1 *consists of at least 3 branches, then the HGT from* (x, y) *to* (z, w)*, transforming* T *into* T_1*, is a part of any minimum cost HGT scenario transforming* T *into* T' *and satisfying the subtree constraint.*

The main steps of the algorithm intended to provide a minimum cost SPR transformation of the given species tree into the given gene tree are the following (the scheme of this algorithm is also presented in Appendix 2).

Preliminary step.—Infer the species and gene trees, denoted respectively by T and T', whose leaves are labeled with the same set of n species. Both trees must be rooted depending on biological evidence. If no plausible evidence for rooting the species and gene trees is available, the outgroup or midpoint strategies can be used to root the trees. The correct tree rooting is essential because a misplaced root in the species or gene tree will lead to false positive and false negative HGTs. If there exist identical subtrees with 2 or more leaves belonging to both T and T', reduce the size of the problem by replacing the identical subtrees by the same single auxiliary branch in both T and T'.

Step k.—Consider all possible HGTs between pairs of branches in the species tree T_{k-1}($T_0 = T$ at Step 1), except the transfers between adjacent branches and those violating the subtree constraint. Among all eligible (i.e., satisfying the subtree constraint) HGTs, look for those satisfying the conditions of Theorem 2 first and Theorem 1 second. Carry out the SPR moves corresponding to these HGTs, thus transforming the tree T_{k-1} into the tree T_k. If no such HGTs exist, carry out all SPR moves corresponding to the transfers satisfying the subtree constraint. Hence, at each step, multiple SPR moves (i.e., multiple HGTs) can be carried out. The direction of each HGT is determined using the selected optimization criterion that can be in our case: LS, RF, QD, or BD. Among 2 opposite HGTs, choose the transfer that minimizes the value of the selected optimization criterion computed for the transformed species tree T_k and gene tree T'. Reduce the size of the problem by collapsing the newly formed subtree(s) in the transformed species tree T_k and the gene tree T'

Stopping Condition, Time Complexity, and Idle Transfer Elimination Procedure

The procedure stops when the RF, LS, QD, or BD coefficient equals zero. Because of the progressive size reduction of the species and gene trees and possibility of identifying multiple HGTs at each step, the time complexity of the proposed algorithm is $O(kn^3)$ to infer that k transfers to reconcile a pair of species and gene phylogenies with n leaves. Once the species and gene trees are reconciled, a backward procedure for eliminating the idle transfers (an idle, or redundant, transfer is the transfer whose removal from the obtained scenario does not change the topology of the resulting gene tree) is carried out. For instance, given the HGT solution shown in Figure 6, the transfer between *Methanococcus jannaschii* and the 5-taxa cluster below including *Archaeoglobus fulgidus*, and performed as HGT number 4, would be an idle transfer (i.e., this HTS would be canceled by SPR moves 4 and 5 presented in Fig. 6). If a k-transfer scenario was found by the algorithm, the backward elimination procedure first tests (k-1) possible subscenarios of HGTs such that in which of them, one of the initially found transfers is eliminated. If no one of the (k-1) subscenarios leads to the same gene tree, then the procedure stops without eliminating HGTs. Otherwise, the first subscenario with (k-1) HGTs that leads to the same gene tree is retained, and all subscenarios with (k-2) HGTs are tested in the similar way. The procedure stops when no more idle HGTs can be found.

Proposition 3 *If the subtree constraint is applied at all steps, then:*

1. *The described HGT detection algorithm has at most $n-3$ steps, and needs at most $n-3$ HGTs (i.e., $n-3$ SPR moves), to transform a binary species tree T with n leaves into a binary gene tree T' with the same set of leaves and*
2. *The gene tree T' is always recovered at the last step of the algorithm (i.e., $T_k = T'$, assuming Step k was the last step of the algorithm) whatever the selected optimization criterion (RF, LS, QD, or BD).*

The proof of this proposition is based on the fact that the maximum number of internal branches in a phylogenetic tree with n leaves is $n-3$ and that each SPR move satisfying the subtree constraint creates at least one new internal branch in the transformed species tree (e.g., branch (x,a) in Fig. 2), existing already in the gene tree T'. Also, whereas the topologies of the transformed species tree and the gene tree T' are different, there exists at least 2 SPR operations (inducing opposite HGTs), satisfying the subtree constraint, that can be carried out. The reader is also referred to Bordewich et al. 2009, theorems 3.1 and 4.1, where the authors prove the existence of a sequence of SPR moves, transforming T into T', in a way that any following tree T_p in the sequence is obtained from T_{p-1} by a single SPR operation and $RF(T_p, T') < RF(T_{p-1}, T')$ (or respectively, $QD(T_p, T') < QD(T_p - 1, T')$). Even though the subtree constraint is not stated in Bordewich et al. (2009), it is implicitly used in the theorem's proofs. The presence of such a sequence of SPR moves is harder to prove theoretically in the case of LS and BD, but the results of a simulation that we conducted for this purpose suggest that it should exist for the 2 latter measures as well.

HGT Bootstrap Validation

Bootstrap analysis is used to place confidence intervals on internal branches of phylogenetic trees (Felsenstein 1985). Here, we extend the HGT bootstrap validation procedure, initially proposed in Makarenkov et al. (2006), to assess the bootstrap support of inferred HGTs. The 3 following strategies can be adopted to evaluate the reliability of the obtained HGTs.

First, the sequence data used to build both species and gene trees are pseudoreplicated. The species and gene trees are inferred from pseudoreplicated sequences by the same tree inferring method used to reconstruct the original species and gene trees. Thus, for all the HGTs being part of the original scenario, we verify if they appear in the HGT scenarios generated with the trees inferred from pseudoreplicates. This verification is carried out by comparing the corresponding SPR moves. In this study, 2 HGTs (or SPR moves) were considered as equal if and only if both donor branch (e.g., branch (x,y) in Fig. 2) and recipient branch (e.g., branch (z,w) in Fig. 2) bipartitions were equivalent in both transfers (i.e., the topologies of the donor and recipient subtrees could be different, but the species content within them was the same in both compared HGTs). An alternative, and stricter, definition would consider that 2 HGTs are equal if and only if the donor and recipient subtrees are identical in both transfers (i.e., the RF distance between them equals 0). Because both species and gene data are pseudoreplicated, such a strategy usually provides low HGT bootstrap scores, especially for badly resolved phylogenies. It is worth noting that not all pseudoreplicated data sets give rise to the species or gene tree whose root branch induces exactly the same root bipartition as that of the original species or gene tree does. If a branch inducing the bipartition identical to the root branch of the reference tree does not exist in the pseudoreplicated tree, then the root of the pseudoreplicated tree can be placed to the branch inducing the closest bipartition, in terms of the Hamming distance, to that induced by the root branch of the corresponding original (species or gene) tree. Such a root positioning strategy is intended to reduce the number of HGTs detected with the pseudoreplicated data (an alternative strategy could utilize an outgroup or a midpoint to root the pseudoreplicated trees).

Second, only the sequence data used to build the gene tree are pseudoreplicated. The sequences used to build the species tree are not resampled. The species tree is taken as an a priori assumption of the method and held constant. In this case, we have to verify that the species tree has a high reliability (e.g., high bootstrap scores). For instance, the species tree can be inferred using appropriate taxonomic information available at the NCBI (The NCBI Handbook 2002) or Tree of Life (Maddison and Schulz 2004) Web sites. The situation when the bipartition corresponding to the root branch of the original gene tree is not found in the tree inferred from pseudoreplicates can be treated in a similar way to the previous case. This bootstrap strategy usually yields higher HGT bootstrap scores than the first one.

Third, HGT bootstrap between 2 tree topologies can be carried out. In contrast to the traditional bootstrap that needs sequence data to compute bootstrap scores, HGT bootstrap can be performed even though only the topologies of the species and gene trees are available. Precisely, we can first execute our program with the exhaustive search option, providing the list of all minimum cost HGT scenarios; this option is also available in the LatTrans program (Hallett and Lagergren 2001). It has an exponential time complexity with respect to the number of HGTs. In our strategy, this option consists of checking all possible SPR moves satisfying the subtree constraint but not only those minimizing at each step the value of the selected optimization criterion. Once the list of all possible minimum cost HGT scenarios is established, we can compute HGT bootstrap scores by estimating the occurrence rate of each HGT in this list.

When the species or gene sequence data are available, the combination of the described strategies (1 and 3) or (2 and 3) can be also carried out to assess HGT bootstrap support. In a general case, Formulas 3 and 4 can be used to compute the bootstrap score *HGT_BS* of the transfer t:

$$HGT\_BS(t) = \left(\sum_{1 \le i \le N_T} \sum_{1 \le j \le N_{T'}} \left(\sum_{1 \le k \le N_{ij}} \frac{\sigma_{k,ij}(t)}{N_{ij}} \times 100\% \right) \right) \Big/ (N_T \times N'_T), \quad \text{and} \tag{3}$$

$$\sigma_{k,ij}(t) = \begin{cases} 1 & \text{if } t \text{ is in the minimum cost scenario } k \text{ for the species tree } T_i \text{ and gene tree } T'_{j'} \\ 0 & \text{if not,} \end{cases} \tag{4}$$

where N_T and N'_T are, respectively, the number of species and gene trees generated from pseudoreplicates and N_{ij} is the number of minimum cost scenarios obtained when carrying out the algorithm with the species tree T_i and gene tree T'_j. The bootstrap score of a HGT scenario can be defined as a product of all individual bootstrap scores found for the transfers being part of this scenario. A comparison of the proposed bootstrap validation technique with the HGT support assessing method included in the "PhyloNet" package (Than et al. 2008a) is provided in the next section.

Simulation Study

Simulation Design

A Monte Carlo study was conducted to test the ability of the new algorithm to recover correct HGTs. Two types of simulations were conducted: In the first one, we considered gene trees with varying confidence levels (their average bootstrap scores ranged from 60% to 100%), whereas in the second, gene trees were assumed not to contain uncertainties and the simulations were carried out with tree-like data only (species trees were assumed to be known in both types of simulations). We examined how the new algorithm performs depending on the selected optimization criterion (including LS, RF, QD, and BD), the number of observed species, and the number of HGTs. Then, the detailed comparison with the LatTrans (Hallett and Lagergren 2001) and RIATA-HGT (Nakhleh et al. 2005; Than and Nakhleh 2008) algorithms was carried out using the optimization strategy based on BD, which yielded the best results among the 4 competing optimization strategies. The simulation procedure included the 4 basic steps described below.

First, a binary species tree T was generated using the random tree generation procedure proposed by Kuhner and Felsenstein (1994). The branch lengths of T were computed using an exponential distribution. Following the approach of Guindon and Gascuel (2002), we added some noise to the branches of the species phylogeny to create a deviation from the molecular clock hypothesis. All branch lengths of T were multiplied by $1 + ax$, where the variable x was obtained from an exponential distribution ($P(x > k) = \exp(-k)$) and the constant a was a tuning factor for the deviation intensity. As in Guindon and Gascuel (2002), the value of a was fixed to 0.8. The random trees generated by this procedure had depth of $O(\log(n))$, where n is the number of species (i.e., number of leaves in a binary phylogenetic tree).

Second, for the first type of simulations only, where the gene tree was supposed to include uncertainties, we used the "SeqGen" program (Rambaut and Grassly 1997) to generate DNA sequences along the branches of the species tree T constructed at the first step. Because SeqGen gives as result only the sequences associated with the tree leaves, we also wrote a program allowing us to identify all the sequences associated with the internal nodes of the species phylogeny. The SeqGen program was used with the HKY model of nucleotide substitution, model of rate heterogeneity assigning different rates to different sites according to a gamma distribution (with the shape parameter equal to 1.0 and TS/TV ratio equal to 2.0). These settings were selected in order to render the simulation parameters similar to those used in the Examples section. The DNA sequences with 100, 500, 1000, 5000, and 10,000 nucleotides were generated.

Third, for each species tree T, we, in turn, generated gene trees with the same number of leaves by performing a fixed number of random SPR moves (representing HGTs) of its subtrees. A model satisfying the evolutionary constraints (Fig. A1) was implemented to generate random HGTs. For each species tree, the gene trees encompassing different numbers of HGTs, varying from 1 to 10, were generated. In the first type of simulations where the sequence data were analyzed, we proceeded as follows: After each SPR operation, we regenerated, using SeqGen, the sequences associated with all the nodes of the subtree being moved. This regeneration started from the root sequence of this subtree, which was set equal to the sequence associated with the internal node, closest to the tree root, of the recipient branch. For each sequence length, different substitution rates were simulated. Various tree heights, obtained by means of the branch lengths adjustment, were considered in order to attain the variations of the substitution rate (see Posada and Crandall 2001 for more detail). These variations led to the gene trees with different average bootstrap scores, ranging from 60% to 100%. For instance, for the gene trees with 50 leaves and DNA sequences with 1000 nucleotides, the average branch length of 4.3 was necessary to obtain the average bootstrap score of 100%, whereas the average bootstrap score of 60% corresponded to a much shorter average branch length, equal to 0.08. To obtain a necessary average branch length of a gene tree, we divided by a predefined constant value all branch lengths of the corresponding species tree, which were computed at Step 1. Using the "Seqboot" program from the PHYLIP package (Felsenstein 1989), we created 100 replicates of each generated data set. The ML trees were then inferred from the original and replicated sequences using the PHYML (Guindon and Gascuel 2003) method. All

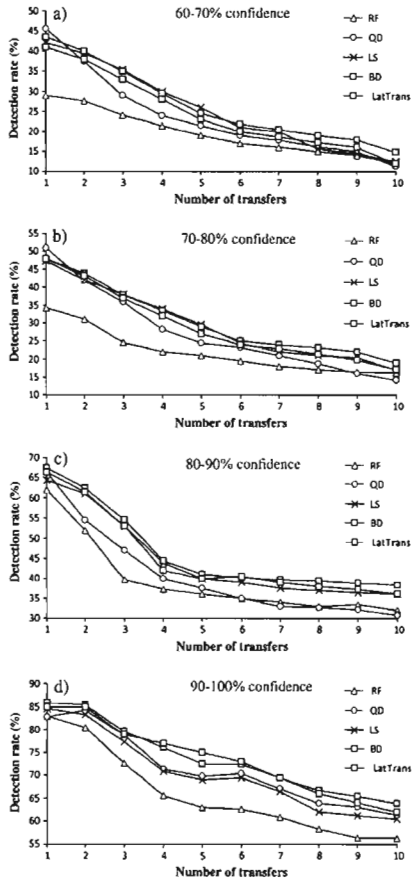


FIGURE 3. Percentage of instances the algorithms recover correct HGTs for the gene trees with the confidence levels: a) 60–70%, b) 70–80%, c) 80–90%, and d) 90–100%. Each reported value represents the average result obtained for random trees with 10, 20, . . . ,100 leaves and DNA sequences with 100, 500, 1000, 5000, and 10,000 nucleotides; 1000 replicates were generated for each combination (tree size, sequence length, and substitution rate) for the LS-, RF-, QD-, and BD-based algorithms and 100 replicates for each combination for LatTrans.

the PHYML parameters were identical to those used in SeqGen. All the phylogenies inferred from DNA sequences were then classified into 1 of the 4 categories (intervals: 60–70%, 70–80%, 80–90%, and 90–100%), depending on their average bootstrap score (the PHYML program also allows users to compute bootstrap scores). The gene trees whose bootstrap scores were lower than 60% were ruled out. A uniform distribution of trees within each of the 4 confidence intervals was attained.

Fourth, the results illustrated in Figures 3, OA5, OA6, OA7 (in online Appendix 2), and 4 were obtained from simulations carried out with random binary phylogenetic trees with $10, 20, \ldots, 100$ leaves. For each tree size, number of HGTs, sequence length, and substitution rate (the last 2 parameters were considered only in the simulations with sequences), 1000 replicated data sets were generated (with an exception of LatTrans in the case of Fig. 3). In the simulations with both sequence and tree-like data, the 4 HGT detection strategies based on LS, RF, QD, and BD, as well as the exhaustive search LatTrans algorithm, were compared (see Figs. 3, OA4, OA5, and OA6). Then, the BD-based strategy was compared (Fig. 4) with the RIATA-HGT algorithm (in the latter case, the comparison was also conducted for nonbinary trees).

Comparison of the LS-, RF-, QD-, and BD-based Algorithms and LatTrans

First, we compared between them the 4 algorithmic strategies discussed in the article in the simulations with sequence data. The behavior of the HGT detection rate versus the number of HGTs is presented in Figure 3. The performances of the LS-, RF-, QD-, and BD-based algorithms, and those of LatTrans, are presented separately for each of the selected confidence intervals of the gene tree (i.e., 60–70%, 70–80%, 80–90%, and 90–100%). The HGT detection rate (i.e., true positives) was measured as a percentage of recovered transfers present in the generated HGT scenario. The BD-based algorithm was generally more accurate than the LS-, RF-, QD-based strategies, and LatTrans in terms of HGT detection rate (Fig. 3). Its performances are more noticeable for the gene trees with higher confidence levels, ranging from 80% to 100% (Fig. 3c,d), when compared with the LS-, RF-, and QD-based strategies, and for the gene trees with lower confidence levels, ranging from 60% to 80% (Fig. 3a,b), when compared with LatTrans. Interestingly, the BD- and LS-based algorithms as well as LatTrans provided very similar results when the number of HGTs was low. For the gene trees with the highest confidence level (Fig. 3d), the BD-based strategy and LatTrans yielded very stable results, which were usually better than those given by the QD- and LS-based algorithms. However, for the gene trees with lower average bootstrap support (Fig. 3a–c), the LS-based strategy usually outperformed its QD and RF counterparts and sometimes showed the results that were very close to those given by the BD-based algorithm and LatTrans. Not surprisingly, the RF-based algorithm was usually worse of the 5 compared techniques regardless of the gene tree confidence level and number of HGTs.

Second, we studied the behavior of the LS-, RF-, QD-, and BD-based algorithms under the conditions of correctness of the gene tree (i.e., tree-like data). Figure OA5a (online Appendix 2) depicts the average HGT detection rates corresponding to the 4 considered

optimization strategies. Figure OA5b depicts the accuracy of the 4 algorithmic strategies in terms of recovery of the complete generated HGT scenario. For both considered criteria (Fig. OA5a,b), the algorithmic strategy based on BD clearly outperformed the strategies based on RF, LS, and QD. The results obtained with QD improve as the number of HGTs increases (Fig. OA5a), and they are only slightly inferior to those obtained with BD as to the identification of the complete HGT scenario (Fig. OA5b). The RF distance was the worse among the 4 competing strategies in terms of both HGT detection rate and identification of total number of transfers. The performances of the BD-based algorithmic strategy were more remarkable in terms of HGT detection rate.

Detailed Comparison with LatTrans

Third, the algorithmic strategy based on BD was compared with the LatTrans algorithm in the case of tree-like data. The comparison of these distance-based algorithms was conducted in terms of HGT detection accuracy and running time. The time complexity of the exhaustive search LatTrans algorithm is $O(2^{\tau}n^2)$, where τ is the number of transfers and n is the number of tree leaves (Hallett and Lagergren 2001). Figure OA6 (a–f in online Appendix 2) depicts the accuracy of both algorithms depending on the number of tree leaves and number of generated transfers. The diagrams in Figure OA6 (a,b) present the true HGT detection rate depending on the number of leaves and generated HGTs. As LatTrans should provide as solution a list of all minimum cost HGT scenarios, we always picked up the first one of the list to compute the LatTrans HGT detection rate (according to Beiko and Hamilton 2006, LatTrans can, however miss some minimum cost HGT scenarios in large phylogenies). Not surprisingly, the detection rate increases as the number of leaves grows. Regarding the detection rate versus number of leaves, LatTrans slightly outperformed the BD-based algorithm (Fig. OA6a) for the trees with 50–70 leaves, whereas our algorithm was better in all other cases. Regarding the detection rate versus number of HGTs (Fig. OA6b), the BD-based algorithm was stronger for big numbers of HGTs (5–10) and weaker for small numbers (1–3). Figure OA6 (c,d) depicts the accuracy of both algorithms when we relax the condition of HGT correctness slightly. Such a relaxed criterion assumes that the algorithm succeeds when it predicts the correct "total number" of HGTs (Hallett et al. 2004). When the total number of HGTs is recovered correctly, the only possibility for not detecting the exact position or direction of some HGTs remains the existence of several minimum or near-minimum cost scenarios (if a near-minimum cost scenario is found). For instance, an opposite direction transfer leading to the same solution (i.e., the same given gene tree) induces a variant of an identical cost scenario (see Maddison 1997 for more details on opposite transfers and Addario-Berry et al. 2003 for a discussion on minimum and near-minimum cost scenarios). It worth noting that sometimes LatTrans generated HGT scenarios not satisfying evolutionary constraints (e.g., in some cases, cyclic HGT scenarios, see Fig. A1(b–d), were found by this method). On average, the BD-based algorithm and LatTrans were able to predict the correct total number of HGTs in 91.1% and 92.5% of cases, respectively (Fig. OA6c,d). We also measured the percentage of instances when the compared algorithms were able to recover a complete generated HGT scenario (Fig. OA6e,f). A complete HGT scenario is recovered if all HGTs found by an algorithm are present in the generated scenario and their total number is also correct. Generally, the BD-based algorithm outperformed LatTrans in terms of complete scenario recovery. This advantage of the BD-based algorithm is mainly due to the presence of HGTs, violating the discussed evolutionary constraints (Fig. A1), in some minimum cost scenarios generated by LatTrans. The polynomial time complexity of our algorithm and the improvement of its results, compared with LatTrans, as the number of leaves or HGTs increases (generally, a slight gain over LatTrans is provided for greater numbers of leaves and HGTs in terms of quality of the obtained transfers) make it particularly interesting for the analysis of large phylogenies encompassing many topological conflicts due to HGT.

Finally, we also compared the running time of the 2 competing algorithms. As previously, the algorithmic performances were assessed with respect to the number of HGTs (Fig. OA7a in online Appendix 2) and number of tree leaves (Fig. OA7b). The simulations were carried out on a PC computer equipped with an Intel Pentium IV dual-core 3.2 GHz processor and 4 GB of RAM. The curves illustrated in Figure OA7 confirm that starting from 30-leave phylogenies and 7 HGTs, our algorithm provides a very significant gain in the running time.

Comparison with RIATA-HGT, HorizStory, and EEEP

In addition to LatTrans (Hallett and Lagergren 2001), which is supposed to infer all possible minimum cost HGT scenarios but is exponential in the number of transfers, various heuristic strategies have been recently developed to detect HGTs. Among the most popular heuristics, we mention HorizStory (MacLeod et al. 2005), EEEP (Beiko and Hamilton 2006), and RIATA-HGT (Nakhleh et al. 2005). All these algorithms are aimed at detecting HGTs by reconciling a given pair of species and gene phylogenies. The PhyloNet package (Than et al. 2008a) includes an extended implementation of the RIATA-HGT algorithm with several improved algorithmic techniques for computing multiple solutions and handling nonbinary trees (Than and Nakhleh 2008). The simulation results presented in Than et al. (2007) and Than and Nakhleh (2008) suggest that the new version of RIATA-HGT significantly outperforms, in terms of speed, the HorizStory, EEEP, and LatTrans algorithms and performs at least as well in terms of accuracy. A new important feature recently added to the PhyloNet package is the estimation of bootstrap support

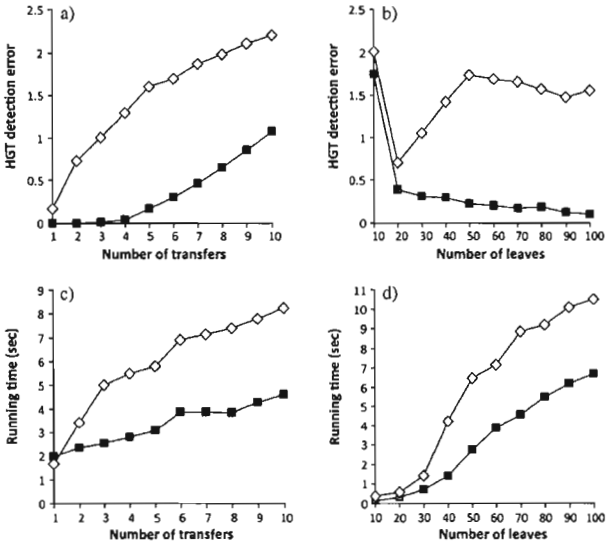


FIGURE 4. HGT detection error consisting of an average absolute difference between the total number of generated and recovered transfers for RIATA-HGT (white diamonds) and the BD-based algorithm (grey squares) depending on the a) number of transfers and b) number of tree leaves. Each reported value represents the combined average result obtained for the set of random binary and nonbinary species trees; 100 binary and 100 nonbinary species trees were generated for each pair of parameters (number of HGTs and tree size). Running time in seconds for the RIATA-HGT and BD-based algorithms depending on the c) number of transfers and d) number of tree leaves.

of HGT branches (Than et al. 2008b). RIATA-HGT does not always recover the minimum cost HGT scenario, but experimental results show very good empirical performance on synthetic and biological data (Nakhleh et al. 2005). It usually generates a multiple set of HGT scenarios of the same length and provides a consensus network for the obtained solutions. On the other hand, the simulation study conducted by Beiko and Hamilton (2006, table 1 and figure 4) to compare the performances of the HorizStory, EEEP, and LatTrans algorithms confirms that LatTrans clearly outperforms HorizStory and EEEP in terms of HGT detection accuracy. For instance, for the trees with 5–20 leaves, the 3 competing methods demonstrated almost perfect HGT recovery (90–100% recovery rates), but for larger trees (30–100 leaves), the performances of HorizStory and EEEP dropped significantly (table 1 in Beiko and Hamilton 2006 reports that for the trees with 100 leaves, the HorizStory average recovery rate is 33.3%, that of EEEP is 70%, and that of LatTrans is 96.7%). Consequently, we decided to compare the proposed BD-based technique to RIATA-HGT (version 1.6), which has a number of common features with our algorithm (e.g., handling nonbinary trees and estimating HGT bootstrap support) and has been the most powerful polynomial-time heuristic in terms of both accuracy and running time.

The comparison with RIATA-HGT was conducted on tree-like data in terms of HGT detection accuracy and running time. Figure 4(a–d) depicts the performances of the RIATA-HGT and BD-based algorithms with respect to the number of tree leaves and generated transfers. The simulations were carried out with both binary and nonbinary trees, and the results presented in Figure 4 are the combined results obtained for both types of trees. First, the species and gene tree data were generated as described above. Second, for the simulation with the nonbinary trees only, some nodes of the binary species trees were merged in order to obtain multifurcations. The number of merging operations was selected randomly and varied from 1 to $n - 3$ for a binary species phylogeny with n leaves. In total, 100 binary and 100 nonbinary species trees were generated for each pair of parameters: Number of HGTs, which ranged from 1 to 10, and Tree size, which ranged from 10 to 100, with a step of 10; gene trees were always binary. The generated benchmark trees used in these simulations can be downloaded from http://www.labunix.uqam.ca/~makarenv/Simulation_trees.zip. Figure 4(a,b) depicts the HGT detection error consisting of an average absolute difference between the total number of generated and recovered transfers. Only nontrivial HGTs were taken into account in these simulations (trivial HGTs, possible in nonbinary trees only, are the transfers between the adjacent branches having in common an internal node of degree bigger than 3; they are only necessary to transform a

nonbinary tree into a binary one). Figure 4 suggests that the BD-based algorithm outperformed RIATA-HGT in terms of combined (binary and nonbinary trees) HGT detection accuracy regardless of the number of leaves and generated transfers. Although the results provided by the 2 algorithms were very similar for binary trees, the BD-based algorithm clearly surpassed RIATA-HGT in the case of nonbinary trees. Moreover, the accuracy of the BD-based algorithm improves as the number of tree leaves grows (Fig. 4b), whereas that of RIATA-HGT remains unstable (mainly due to its bad performance in the case of nonbinary trees). In terms of running time, the advantage also goes to the BD-based algorithm regardless of the number of leaves and generated transfers (Fig. 4c,d). The comparison of the results provided by the RIATA-HGT and BD-based algorithms for both real data sets considered in this article is made in the Examples section.

Than et al. (2008b) also proposed a method, now included in the PhyloNet package, for assessing the support of HGT branches. Figure OA8 (online Appendix 3) presents an illustration of computing the support value of a HGT branch by RIATA-HGT (see also fig. 8 in Than et al. 2008b). In the latter study, the support of the HGT branch $X \rightarrow Y$ added to the species tree is defined as the maximum bootstrap support of all internal branches of the path linking the nodes Z and X in the gene tree. The bootstrap support of the event $X \rightarrow Y$ given by RIATA-HGT in this case is 100%, disregarding the low bootstrap support of 10% of the internal branch separating the leaves B and D from the rest of the tree. We think that the bootstrap scores of HGT events computed in this way are largely overestimated. Furthermore, this way of assessing the HGT bootstrap support does not take into account the topologies of the replicated gene phylogenies (the species phylogeny is assumed to be fixed). A unique gene tree with the given bootstrap scores of its internal branches does not always encompass all important features of the set of replicated trees that were used to calculate these scores. Even though the bootstrap support of each clade is indicated in such a unique gene tree, the key information, concerning the percentage of occurrences when 2 clades affected by a HGT event are present together in the replicated gene trees, is missing. In our method, each replicated tree is tested in turn, and the obtained HGT statistics are combined (see Formulas 3 and 4) to calculate HGT bootstrap support. For instance, the bootstrap score of the HGT event $X \rightarrow Y$ (Fig. OA8) computed by our method would be at most 10%.

EXAMPLES

Detecting Horizontal Transfers of the Gene rpl12e

We first examined the evolution of the gene *rpl12e* for the 14 organisms of archaea originally considered by Matte-Tailliez et al. (2002). The latter authors discussed the problems encountered when reconstructing some parts of the archaeal phylogeny and pointed out the evidence of HGT events influencing the evolution of *rpl12e*. Matte-Tailliez et al. (2002) inferred the ML tree of the gene *rpl12e* (Fig. 5) for 14 organisms of archaea and compared it with the ML phylogeny (Fig. 6,

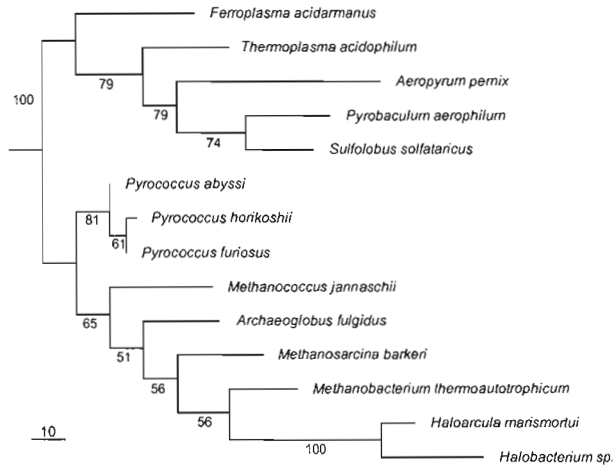


FIGURE 5. ML phylogenetic tree for the protein *rpl12e* (89 positions). Numbers close to branches are the ML bootstrap scores obtained from the sampled protein sequences using the Seqboot and Proml (JTT model, Jones et al. 1992) programs from the PHYLIP package (Felsenstein 1989). The tree topology is identical to that found by Matte-Tailliez et al. (2002, fig. 3).

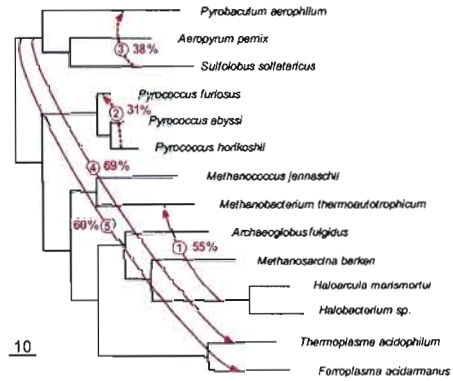


FIGURE 6. Species tree (Matte-Tailliez et al. 2002, fig. 1a) with 5 HGTs indicated by arrows. Numbers on HGTs indicate their order of inference. HGT bootstrap scores are indicated near to the numbers of the corresponding HGTs. Arrows 4 and 5 depict the HGTs between the clades of Thermoplasmatales and Crenarchaeota originally predicted by Matte-Tailliez et al. (2002). HGTs with bootstrap scores of 50% or less are depicted by dashed arrows.

undirected lines) based on the concatenated 53 ribosomal proteins (7175 positions). Calculation of the α parameter values and other ML analyses, taking into account among-site rate variation and Γ-law correction, for the 53 concatenated proteins were carried out by Matte-Tailliez et al. (2002) using the PUZZLE program (Strimmer and von Haeseler 1996). Given the topological incongruence of the obtained phylogenies, the authors hypothesized a few cases of HGT of the gene *rpl12e*. More precisely, the case of the HGT between the clades of Thermoplasmatales (*Ferroplasma acidarmanus* and *Thermoplasma acidophilum*) and Crenarchaeota (*Aeropyrum pernix, Pyrobaculum aerophilum*, and *Sulfolobus solfataricus*) was indicated as the most evident one.

We first reconstructed from the original sequences the topologies of the gene (Fig. 5) and species trees (Fig. 6, undirected lines). The HGT detection was performed with the algorithmic strategy based on the BD. Five transfers needed to reconcile the species and gene topologies were found (they are indicated by arrows in Fig. 6). The transfer between the clade of *Halobacterium* sp. and *Haloarcula marismortui* and *Methanobacterium thermoautotrophicum* was found in the first step. Its bootstrap support, computed by fixing the topology of the species tree and replicating the gene tree sequences, is 55%.

In the second and third steps, we found the HGTs between *Pyrococcus horikoshii* and *P. furiosus* (Step 2) and between *S. solfataricus* and *P. aerophilum* (Step 3). Both these HGTs link closely related species and have low bootstrap scores of 31% and 38%, respectively. The low bootstrap scores of these HGTs can be explained by the possibility of the opposite HGTs leading, in both cases, to the same topological rearrangements as those induced by the obtained transfers.

The HGTs 4 and 5 link the clade of Crenarchaeota to the organisms *T. acidophilum* and *F. acidarmanus*. The transfers between these 2 groups were also predicted by Matte-Tailliez et al. (2002). The identical direction and similar bootstrap scores of the HGTs 4 and 5 suggest that a unique HGT, instead of these 2 transfers, might take place between the clades of Thermoplasmatales and Crenarchaeota. It is worth noting that any algorithm based on the minimization of the SPR distance would find 2 transfers in this case. An intuitively unique HGT linking these clades was disguised most likely as a result of an artifact affecting the reconstruction of the gene tree (Fig. 5). For instance, if the organisms *T. acidophilum* and *F. acidarmanus* were neighbors (i.e., the leaves corresponding to these organisms were incident with same internal vertex) in the gene tree, a unique HGT from the Crenarchaeota clade to the Thermoplasmatales clade, instead of HGTs 4 and 5 presented in Figure 6, would be sufficient to recover the correct topology of the gene tree.

In total, 4 minimum cost HGT scenarios were found for the considered species and gene trees. All of them include HGTs 1, 4, and 5. However, the HGTs 2 and 3 can be as presented in Figure 6 or go to the opposite direction; this accounts for their low bootstrap scores computed using Formulas 3 and 4.

For the example of the *rpl12e* data, RIATA-HGT found 9 solutions, each of size 5 (Fig. OA9; online Appendix 3 includes the input data and exact output data provided by RIATA-HGT). Five of these solutions contradict the same lineage constraint (they include a HGT marked by [time violation?] in the program output), and 4 of them satisfy all plausible evolutionary constraints. The solution represented in Figure 6 is among those 4 eligible solutions. The HGT bootstrap scores found by RIATA-HGT are indicated between the parentheses in

the program output (online Appendix 3). They are generally much higher than the corresponding bootstrap scores calculated by our method. For instance, the perfect 100% scores for the HGTs 4 and 5 (Fig. 6) were found by RIATA-HGT, despite the 79% score of the gene tree branch (Fig. 5) linking *T. acidophilum* and the clade of Crenarchaeota.

Detecting Horizontal Transfers of PheRS Synthetase

Woese et al. (2000) analyzed from the evolutionary perspective the relationship of the aminoacyl-tRNA synthetases (AARSs) to their genetic code. They found that the AARSs are very informative about the evolutionary process. Analysis of different phylogenetic trees for a number of considered AARSs revealed the following features: The AARSs evolutionary relationships are mostly conform to established organismal (i.e., species) phylogeny; a strong distinction exists between bacterial and archaeal types of AARSs; horizontal transfer of AARS genes between bacteria and archaea is asymmetric: HGT of archaeal AARSs to the bacteria is more prevalent than the reverse.

We examined the evolution of the PheRS sequences for the set of 32 organisms considered by Woese et al. (2000, fig. 2), including 24 bacteria, 6 archaea and 2 eukarya. As suggested by the latter authors, it is tempting to view the evolution of aminoacyl-tRNA synthesis from top to bottom as a HGT study. The PheRS phylogenetic tree inferred with PHYML (Guindon and Gascuel 2003) is shown in Figure 7. This tree is slightly different from that obtained by Woese et al. (2000, fig. 2). The biggest difference consists of the presence in the phylogeny in Figure 7 of a new clade formed by 2 eukarya (*Homo sapiens* and *Saccharomyces cerevisiae*) and 2 archaea (*A. fulgidus* and *M. thermoautotrophicum*). This 4-taxa clade, not appearing in the consensus tree (not shown here), has a low bootstrap support and is probably due to tree reconstruction artifacts.

PheRS is the only class II synthetase in the NUN codon group, and it has no close relatives within that class. For both the α- and the β-subunits of PheRS,

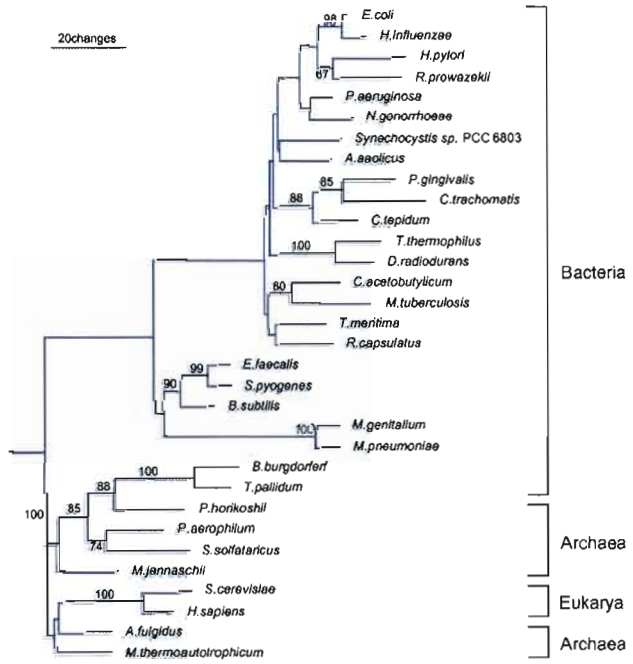


FIGURE 7. Phylogenetic tree of PheRS sequences. Protein sequences with 171 bases were aligned using ClustalW (Thompson et al. 1994). Additional alignment optimization was performed with MUST (Philippe 1993). Badly aligned regions were removed using GBlocks (Castresana 2000); 160 bases were conserved. The ML tree was then inferred with PHYML (Guindon and Gascuel 2003) using Γ-law correction. Bootstrap scores above 60% are indicated. Tree was rooted between the bacteria and the archaea plus eukarya. The sequence identifiers correspond to organisms reported in table 2 of Woese et al. (2000).

significant length differences distinguish the bacterial subunits from their archaeal counterparts (Woese et al. 2000). PheRS shows the classical canonical pattern, the only exception being the spirochete (i.e., *Borrelia burgdorferi and Treponema pallidum*) PheRSs. They are of the archaeal, not the bacterial, genre and seem to be specifically related to *P. horikoshii* within that grouping (see fig. 7 or fig. 2 in Woese et al. 2000). The sequence signature analysis confirms this fact.

The species phylogeny corresponding to the NCBI (The NCBI Handbook 2002) taxonomic classification was also constructed (Fig. 8, undirected lines). Note that in this case, the species phylogeny is not a fully resolved tree; it contains 5 internal nodes of degree bigger than 3. The 7 nontrivial HGTs (see the previous section for the definition of a trivial transfer) with their bootstrap scores found by our algorithm are shown in Figure 8. In total, the algorithm found 17 HGTs including 10 trivial transfers that are not presented here. The transfer number 6, having the bootstrap support of 86%, links the organism *P. horikoshii* and the clade of spirochetes, including *B. burgdorferi* and *T. pallidum*. This bootstrap score is very close to the biggest possible score of 88% that could be obtained for this HGT (see the corresponding 3-taxa clade in the PheRS phylogeny shown in Fig. 7). This transfer confirms the hypothesis that the PheRS gene of spirochetes was involved in HGT. On the other hand, the low HGT bootstrap scores of the 3 nontrivial HGTs (1, 3, and 5 shown by dashed arrows in Fig. 8) can be explained by weak bootstrap support of the related branches in the gene phylogeny (Fig. 7). For instance, the HGT number 1 linking the archaea *A. fulgidus* to the clade of 2 eukarya has the lowest bootstrap score of 25% only. In this example, the solution found using BD as an optimization criterion is shown. The use of the RF, QD, or LS optimization, instead of BD, leads to the same HGT scenario differing from that shown in Figure 8 only by the HGT bootstrap scores. For these data, a unique minimum cost HGT scenario with 7 nontrivial transfers was found by the new algorithm. Note that this data set was originally analyzed in Makarenkov et al. (2006) using a "greedy" HGT detection algorithm based on the RF (and LS) optimization. The solution found in the 2006 paper (see fig. 5, page 347), using both RF and LS, consisted of 9 nontrivial HGTs needed to transform the nonbinary species tree in Figure 8 (undirected lines) into the binary gene tree in Figure 7. In this example, the use of the new algorithm allowed us to obtain a unique minimum cost HGT scenario consisting of 7 nontrivial transfers only (e.g., the HGT from *Helicobacter pylori* and *Rickettsia prowazekii* shown in fig. 5 in Makarenkov et al. 2006 is not a part of the optimal HGT scenario presented in Fig. 8).

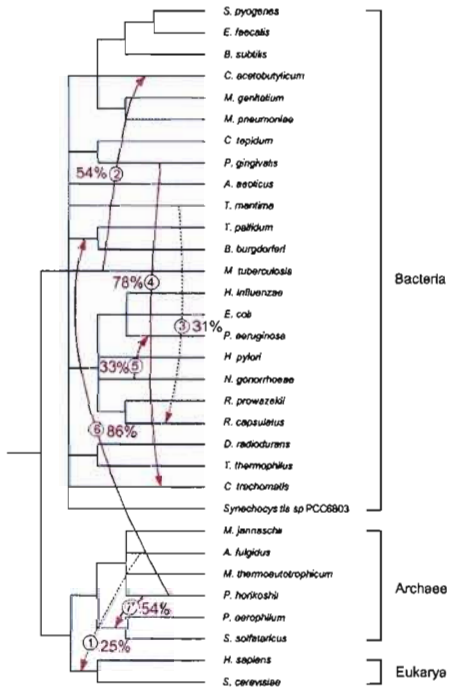


FIGURE 8. Nonbinary species phylogeny (undirected lines) corresponding to the NCBI taxonomic classification for the 32 organisms from Figure 7. The 7 nontrivial HGTs (indicated by arrows), including 4 HGTs with bootstrap scores above 50% (solid arrows) and 3 HGTs with bootstrap score lower than 50% (dashed arrows) were found. HGT bootstrap scores are indicated near to the numbers of the corresponding HGTs.

For these data, RIATA-HGT found 12 solutions, each of them of size 14, including nontrivial transfers only (see Fig. OA10 in online Appendix 3). Five initial species tree transformations indicated by the dashed ellipses in Figure OA10 were made by RIATA-HGT prior to carrying out HGT detection. Each of these transformations corresponds to a trivial HGT. Thus, the solution presented in Figure OA10 actually consists of 19 HGTs, comprising 14 regular and 5 trivial HGTs. The minimum cost solution found by the BD-based algorithm, and consisting of 7 regular and 10 trivial HGTs, was not found by RIATA-HGT. As in the previous example, the HGT bootstrap scores found by RIATA-HGT were generally much higher than those found by our algorithm (see the RIATA-HGT output in online Appendix 3). For instance, a perfect score of 100% was found by RIATA-HGT for the HGT stemming from the archaebacterium *P. horikoshii* and going to the cluster of spirochetes (HGT number 6 in Fig. 8), whereas the bootstrap score of the clade regrouping these organisms in the gene tree is 88% (Fig. 7).

DISCUSSION

HGT is one of the main mechanisms contributing to microbial genome diversification. It is rampant among various groups of genes in bacteria (Doolittle 1999). For instance, over the long term, it may be the dominant force, affecting most genes in most prokaryotes (Doolittle et al. 2003). At the same time, HGT poses several risks to humans, including antibiotic-resistant genes spreading to pathogenic bacteria, transgenic DNA inserting into human cell and triggering cancer, and disease-associated genes spreading and recombining to create new viruses and bacteria (Nakhleh et al. 2005). In this article, we described an accurate polynomial-time algorithm for inferring HGT events. Each HGT mapped into the species phylogeny aids to reconcile the topologies of the species and gene trees. Both species and gene trees can be inferred from the sequence or distance data, and both can include uncertainties. The presented algorithm can rely either on the metric, using LS, or on the topological optimization, using the RF distance, QD or BD, to predict HGT events. The BD measure introduced in this article can be viewed as an interesting refinement of the RF metric. It allows for capturing the degree of dissimilarity of unequal subtrees, what the widely used RF distance fails to achieve. According to the simulation results, the BD, intended to compare the "quality" of the tree bipartitions, and not their "quantity" as the RF metric does, is much more appropriate than RF for finding optimal scenarios of SPR moves (i.e., HGTs) for the given pair of species and gene phylogenies (see the example of the "caterpillar-shaped" tree in Fig. OA4 in online Appendix 1).

The discussed algorithm has a number of important properties and advantages. First, Theorems 1 and 2, used in the algorithmic procedure, enable one to infer transfers being part of any (or of some) minimum cost HGT scenario(s). The described algorithm is not limited to binary species trees. The example of the PheRS data confirms that it can be used in the case when the species tree is not fully resolved. In this case, trivial HGTs will be produced by the algorithm. They should be ignored in the final solution. On the other hand, the case where the considered species and gene trees have different numbers of leaves could be also handled by the new algorithm. In this situation, we have first to find the maximum subset of identical species (i.e., leaves) present in both trees and then repeatedly collapse, in both of them, all branches connected to the species not included in this subset until the trees comprise identical sets of leaves. Once the collapsing operation is over, the method can be applied as described. Also, the situation where more than one copy of a gene is considered could be handled by introducing auxiliary species in the species tree, each of them representing a different copy of the gene. Both latter cases constitute a promising direction for further research.

Furthermore, the considered subtree constraint (Fig. 2) offers a number of important advantages. First, the order of HGTs inferred under this constraint is opposite to their real evolutionary order. Most of the HGT detection programs (e.g., LatTrans) do not provide HGTs in the strict evolutionary order. Second, it takes care of all necessary evolutionary constraints (Fig. A1; see also Maddison 1997 or Page and Charleston 1998), such as the transfers within the same lineage or some crossing transfers. All these constraints are taken into account automatically while using the subtree constraint because both subtrees involved in the HGT have to be present in the gene tree as well as the new subtree that they form after the transfer (Fig. 2). Third, the use of this constraint allows us to reduce the size of the problem at each step of the algorithm by collapsing the identical subtrees in both species and gene phylogenies and replacing them by single auxiliary branches. Fourth, the 2 last arguments also offer an important gain in running time for this problem known to be computationally hard. The importance of such a gain shows off particularly when carrying out HGT bootstrap validation.

As any method of phylogenetic analysis, the described HGT detection algorithm is subject to a number of artifacts that generally affect phylogenetic inferring, the main of them being long-branch attraction, unequal evolutionary rates, and situations when some HGT events almost coincides with some speciation events. In the future, it will be important to investigate in greater detail the impact of these artifacts on the performances of HGT detection algorithms. In some cases, the described algorithm may fail to obtain a correct HGT scenario or may infer HGTs going to the opposite direction. The latter case appears when a couple of HGTs that differ only by their direction lead to the same topological rearrangement of the species tree (e.g., HGTs 2 and 3 in Fig. 6). Such transfers usually have low bootstrap support. The issue of noninferring a correct HGT scenario is characteristic of small trees encompassing high number of transfers. However, the exhaustive search LatTrans algorithm (Fig. OA6 in online Appendix 2) and the RIATA-HGT heuristic (Fig. 4) also do not cope well with these situations (our algorithm usually outperformed both of them under these conditions).

A comprehensive simulation study was conducted in order to compare the 4 considered measures (LS, QD, RF, and BD) in the context of HGT inferring. The simulations demonstrated that the BD-based algorithm outperformed those based on the LS, QD, and RF criteria in most circumstances (Figs. 3 and OA5 in online Appendix 2). The RF-based procedure proved to be the less reliable among the 4 strategies. Then, the BD-based procedure was compared with the exact exponential-time algorithm LatTrans (Hallett and Lagergren 2001) and to a fast and accurate heuristic RIATA-HGT (Nakhleh et al. 2005; Than and Nakhleh 2008) in terms of both accuracy of HGT recovery and running time. Although the new algorithm and LatTrans yielded very similar results in terms of HGT recovery (Figs. 3 and OA6), our algorithm remained much faster than LatTrans (Fig. OA7). On the other hand, the BD-based strategy outperformed RIATA-HGT in terms of both HGT detection error and

running time (Fig. 4) in a combined simulation study carried out for binary and nonbinary phylogenies.

Mention that the new algorithm can be particularly useful when validating HGTs by bootstrap. Three ways of carrying out HGT bootstrap validation were suggested depending on the data at hand. The computation of HGT bootstrap support can be carried out taking into account the robustness of the species tree, that of the gene tree, and the ratio of the obtained HGTs in all minimum cost scenarios found for the given pair of species and gene trees (Formulas 3 and 4).

The new version of the "T-Rex" program (Makarenkov 2001) including the described algorithm for predicting and validating HGT events and the input data for the discussed *rpl12e* and *PheRS* synthetase examples are freely available at the following URL: http://www.trex.uqam.ca.

Supplementary Material

Supplementary material can be found at http://www.sysbio.oxfordjournals.org/.

Funding

This research was supported by the Natural Sciences and Engineering Research Council of Canada.

Acknowledgements

We are grateful to Drs Olivier Gascuel, Jack Sullivan, Christophe Dessimoz, and 2 anonymous reviewers for their helpful comments and discussions.

References

Acinas S.G., Marcelino L.A., Klepac-Ceraj V., Polz M.F. 2004. Divergence and redundancy of 16S rRNA sequences in genomes with multiple rrn operons. J. Bacteriol. 38:2629–2635.

Addario-Berry L., Hallett M., Lagergren J. 2003. Towards identifying lateral gene transfer events. Pac. Symp. Biocomput. 8:279–290.

Allen B.L., Steel M. 2001. Subtree transfer operations and their induced metrics on evolutionary trees. Ann. Combin. 5:1–15.

Beiko R.G., Hamilton N. 2006. Phylogenetic identification of lateral genetic transfer events. BMC Evol. Biol. 6:15.

Boc A., Makarenkov V. 2003. New efficient algorithm for detection of horizontal gene transfer events. In: Benson G., Page R., editors. Algorithms in bioinformatics. Berlin/Heidelberg (Germany): Springer Verlag. p. 190–201.

Bordewich M., Gascuel O., Huber K.T., Moulton V. 2009. Consistency of topological moves based on the balanced minimum evolution principle of phylogenetic inference. IEEE/ACM Trans. Comput. Biol. Bioinform. 6:110–117.

Bordewich M., Semple C. 2004. On the computational complexity of the rooted subtree prune and regraft distance. Ann. Combin. 8:409–423.

Castresana J. 2000. Selection of conserved blocks from multiple alignments for their use in phylogenetic analysis. Mol. Biol. Evol. 17:540–552.

Charleston M.A. 1998. Jungle: a new solution to the host/parasite phylogeny reconciliation problem. Math. Biosci. 149:191–223.

Csürös M., Miklós I. 2006. A probabilistic model for gene content evolution with duplication, loss, and horizontal transfer. In: Istrail S., Pevzner P., Waterman M., editors. Research in computational molecular biology. Lecture Notes in Computer Science. Berlin/Heidelberg (Germany): Springer Verlag.

Doolittle W.F. 1999. Phylogenetic classification and the universal tree. Science. 284:2124–2129.

Doolittle W.F., Boucher Y., Nesbo C.L., Douady C.J., Andersson J.O., Roger A.J. 2003. How big is the iceberg of which organellar genes in nuclear genomes are but the tip? Philos. Trans. R. Soc. Lond. B Biol. Sci. 358:39–57.

Felsenstein J. 1985. Confidence limits on phylogenies: an approach using the bootstrap. Evolution. 39:738–791.

Felsenstein J. 1989. PHYLIP: phylogeny inference package. Version 3.2. Cladistics. 5:164–166.

Gogarten J.P., Doolittle W.F., Lawrence J.G. 2002. Prokaryotic evolution in light of gene transfer. Mol. Biol. Evol. 19:2226–2238.

Guindon S., Gascuel O. 2002. Efficient biased estimation of evolutionary distances when substitution rates vary across sites. Mol. Biol. Evol. 19:534–543.

Guindon S., Gascuel O. 2003. A simple, fast and accurate algorithm to estimate large phylogenies by maximum likelihood. Syst. Biol. 52:696–704.

Hallett M., Lagergren J. 2001. Efficient algorithms for lateral gene transfer problems. In: El-Mabrouk N., Lengauer T., Sankoff D., editors. Proceedings of the Fifth Annual International Conference on Research in Computational Biology. New York: ACM Press. p. 149–156.

Hallett M., Lagergren J., Tofigh A. 2004. Simultaneous identification of duplications and lateral transfers. In: Bourne P.E., Gusfield D., editors. In: Proceedings of the Eighth Annual International Conference on Research in Computational Biology. San Diego (CA): ACM. p. 347–356.

Hein J. 1990. A heuristic method to reconstructing the evolution of sequences subject to recombination using parsimony. Math. Biosci. 98:185–200.

Hein J., Jiang T., Wang L., Zhang K. 1996. On the complexity of comparing evolutionary trees. Discrete Appl. Math. 71:153–169.

Hickey G., Dehne F., Rau-Chaplin A., Blouin C. 2008. SPR distance computation for *unrooted* trees. Evol. Bioinform. Online 4:17–27.

Jin G., Nakhleh L., Snir S., Tuller T. 2006. Efficient parsimony-based methods for phylogenetic network reconstruction. Bioinformatics. 23:123–128.

Jin G., Nakhleh L., Snir S., Tuller T. 2007. Inferring phylogenetic networks by the maximum parsimony criterion. Mol. Biol. Evol. 24(1):324–337.

Jones D.T., Taylor W.R., Thornton J.M. 1992. The rapid generation of mutation data matrices from protein sequences. CABIOS 8:275–282.

Koonin E.V. 2003. Horizontal gene transfer: the path to maturity. Mol. Microbiol. 50:725–727.

Kuhner M., Felsenstein J. 1994. A simulation comparison of phylogeny algorithms under equal and unequal evolutionary rates. Mol. Biol. Evol 11:459–468.

Lawrence J.G., Ochman H. 1997. Amelioration of bacterial genomes: rates of change and exchange. J. Mol. Evol. 44:383–397.

MacLeod D., Charlebois R.L., Doolittle F, Bapteste E. 2005. Deduction of probable events of lateral gene transfer through comparison of phylogenetic trees by recursive consolidation and rearrangement. BMC Evol. Biol. 5:27.

Maddison W.P. 1997. Gene trees in species trees. Syst. Biol. 46:523–536.

Maddison D.R., Schulz K.S. 2004. The Tree of Life web project [Internet]. Available from: http://tolweb.org.

Makarenkov V. 2001. T-Rex: reconstructing and visualizing phylogenetic trees and reticulation networks. Bioinformatics. 17:664–668.

Makarenkov V., Boc A., Delwiche C.F., Diallo A.B., Philippe H. 2006. New efficient algorithm for modeling partial and complete gene transfer scenarios. In: Batagelj V., Bock H.-H., Ferligoj A., Ziberna A., editors. Data science and classification. Berlin/Heidelberg (Germany): Springer Verlag.

Matte-Tailliez O., Brochier C., Forterre P., Philippe H. 2002. Archaeal phylogeny based on ribosomal proteins. Mol. Biol. Evol. 19:631–639.

Mirkin B.G., Fenner T.I., Galperin M.Y., Koonin E.V. 2003. Algorithms for computing parsimonious evolutionary scenarios for genome evolution, the last universal common ancestor and dominance of horizontal gene transfer in the evolution of prokaryotes. BMC Evol. Biol. 3:2.

Mirkin B.G., Muchnik I., Smith T.F. 1995. A biologically consistent model for comparing molecular phylogenies. J. Comput. Biol. 2: 493–507.

Moret B.M.E., Nakhleh L., Warnow T., Linder C.R., Tholse A., Padolina A., Sun J., Timme R.E. 2004. Phylogenetic networks: modeling, reconstructibility and accuracy. IEEE/ACM Trans. Comput. Biol. Bioinform. 1:13–23.

Nakhleh L., Ruths D., Wang L. 2005. RIATA-HGT: a fast and accurate heuristic for reconstructing horizontal gene transfer. In: Lusheng Wang, editor. Computing and Combinatorics, 11th Annual International Conference, COCOON 2005, Kunming, China, August 16–29, 2005, Proceedings. Lecture Notes in Computer Science 3595 Springer 2005, ISBN 3-540-28061-8.

Page R.D.M. 1994. Maps between trees and cladistic analysis of historical associations among genes, organism and areas. Syst. Biol. 43:58–77.

Page R.D.M., Charleston M.A. 1998. Trees within trees: phylogeny and historical associations. Trends Ecol. Evol. 13:356–359.

Philippe H. 1993. MUST: a computer package of management utilities for sequences and trees. Nucl. Acids Res. 21:5264–5272.

Posada D., Crandall K.A. 2001. Selecting the best-fit model of nucleotide substitution. Syst. Biol. 50:580–601.

Rambaut A., Grassly N.C. 1997. Seq-Gen: an application for the Monte Carlo simulation of DNA sequence evolution along phylogenetic trees. Comput. Appl. Biosci. 13:235–238.

Robinson D.R., Foulds L.R. 1981. Comparison of phylogenetic trees. Math. Biosci. 53:131–147.

Strimmer K., von Haeseler A. 1996. Quartet puzzling: a quartet maximum likelihood method for reconstructing tree topologies. Mol. Biol. Evol. 13:964–969.

Than C., Jin G., Nakhleh L. 2008a. Integrating sequence and topology for efficient and accurate detection of horizontal gene transfer. In: Nelson C.E., Stéphane Vialette, editors. Comparative Genomics, International Workshop, RECOMB-CG 2008, Paris, France, October 13–15, 2008. Proceedings. Lecture Notes in Computer Science 5267. Springer Berlin/Heidelberg, ISBN 978-3-540-87988-6.

Than C., Nakhleh L. 2008. SPR-based tree reconciliation: non-binary trees and multiple solutions. In: Alvis Brazma, Satoru Miyano, Tatsuya Akutsu, editors. Proceedings of the 6th Asia-Pacific Bioinformatics Conference, APBC 2008, 14–17 January 2008, Kyoto, Japan. Advances in Bioinformatics and Computational Biology 6 Imperial College Press 2008, ISBN 978-1-84816-108-5. p. 251–260.

Than C., Ruths D., Innan H., Nakhleh L. 2007. Confounding factors in HGT detection: statistical error, coalescent effects, and multiple solutions. J. Comput. Biol. 14:517–535.

Than C., Ruths D., Nakhleh L. 2008b. PhyloNet: a software package for analyzing and reconstructing reticulate evolutionary relationships. BMC Bioinform. 9:322.

The NCBI Handbook. 2002. Chapter 17, The Reference Sequence (RefSeq) Project [Internet]. Bethesda (MD): National Library of Medicine (US), National Center for Biotechnology Information.

Thompson J.D., Higgins D.G., Gibson T.J. 1994. CLUSTAL W: improving the sensitivity of multiple sequence alignment through sequence weighting, positions-specific gap penalties and weight matrix choice. Nucl. Acids Res. 22:4673–4680.

Tsirigos A., Rigoutsos I. 2005. A new computational method for the detection of horizontal gene transfer events. Nucl. Acids Res. 33: 922–933.

von Haeseler A., Churchill G.A. 1993. Network models for sequence evolution. J. Mol. Evol. 37:77–85.

Woese C.R., Olsen G., Ibba M., Söll D. 2000. Aminoacyl-tRNA synthetases, the genetic code, and the evolutionary process. Microbiol. Mol. Biol. Rev. 64:202–236.

Zhaxybayeva O., Lapierre P., Gogarten J.P. 2004. Genome mosaicism and organismal lineages. Trends Genet. 20:254–260.

APPENDIX 1

This Appendix includes an illustration of required evolutionary constraints and an illustration for Theorems 1 and 2.

Evolutionary constraints

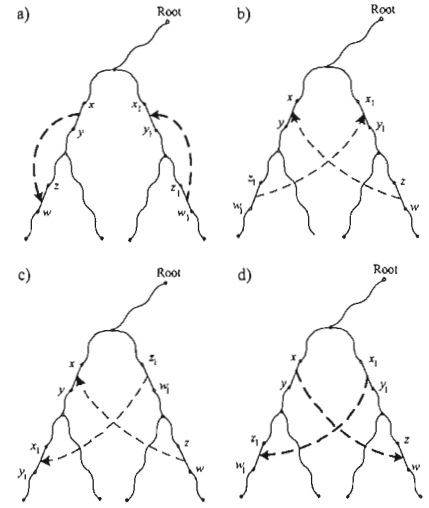


FIGURE A1. HGTs between the branches located on the same lineage (case a) should be prohibited. HGTs crossing in these ways (cases b, c and d) should be prohibited. A single tree branch is depicted by a plane line and a path is depicted by a wavy line.

Illustration for Theorems 1 and 2

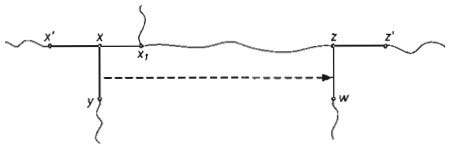


FIGURE A2. HGT from the branch (x, y) to the branch (z, w) is a part of (Theorem 1): a minimum cost HGT scenario transforming the species tree T into the gene tree T' if the bipartition corresponding to the branch (x, x_1) in the transformed species tree T_1 is present in the bipartition table of T' and the subtree Sub_{yw} (i.e., obtained by the SPR move induced by this HGT, see Fig. 2) is present in the tree T'; (Theorem 2): any minimum cost HGT scenario transforming the species tree T into the gene tree T' if all the bipartitions corresponding to the branches of the path (x', z') in the transformed species tree T_1 are present in the bipartition table of T' and the subtree Sub_{yw} is present in the tree T'. A single tree branch is depicted by a plane line and a path is depicted by a wavy line.

APPENDIX 2

This Appendix includes the scheme of the described heuristic algorithm for finding a minimum-cost SPR transformation of the given species tree into the given gene tree.

Infer species and gene trees T and T' on the same set of species (i.e., leaves);

Root T and T' according to biological evidence or using an outgroup or a midpoint;

if (there exist identical subtrees with two or more leaves in T and T') **then**
Decrease the size of the problem by collapsing them in both T and T';

Select the optimisation criterion OC = LS (least-squares), or RF (Robinson and Foulds distance), or QD (quartet distance), or BD (bipartition dissimilarity);

Compute the initial value of OC between T and T';
$T_0 = T$;
$k = 1$; //k is the Step index

while (OC $\neq 0$)
{
Find the set of all eligible HGTs (i.e., SPR moves) at step k (denoted by E_HGT_k);
The set E_HGT_k contains only the transfers satisfying *the subtree constraint*;
while (*HGTs satisfying the conditions of Theorems 2 and 1 exist*)
{
if (there exist HGTs $\in E\_HGT_k$ and satisfying the conditions of Theorems 2) **then**
Carry out the SPR moves corresponding to these HGTs;

if (there exist HGTs $\in E\_HGT_k$ and satisfying the conditions of Theorem 1) **then**
Carry out the SPR moves corresponding to these HGTs;
}
Carry out *all remaining SPR moves* corresponding to HGTs satisfying the subtree constraint;
Compute the value of OC to identify the direction of each HGT;

$k = k + 1$;
Decrease the size of the problem by collapsing the identical subtrees in T_k and T';
Compute the value of OC between T_k and T';
}

Eliminate the idle transfers from the obtained scenario using a backward elimination procedure;

end.

ONLINE APPENDIX 1

This Appendix includes Propositions 1 and 2, Theorems 1, 2 and 3 with their proofs as well as an example showing inappropriateness of the RF metric in the HGT recovery context.

Properties of the Bipartition Dissimilarity

The Propositions 1 and 2 establish some interesting properties of the bipartition dissimilarity. Thus, Proposition 1 states the sufficiency condition that ensures that a bipartition dissimilarity (BD) satisfies the triangle inequality (and is a metric), and Proposition 2 gives the maximum values of this measure depending on the number of tree leaves.

The bipartition a of a tree T *is associated* to the bipartition b of the tree T' (this association is denoted by $a \rightarrow b$), if the Hamming distance between the bipartition vectors corresponding to a and b is the smallest among all possible distances computed between a and all the bipartition vectors corresponding to the branches of the tree T'. A sufficient metricity condition is as follows:

Proposition 1. *Let T_1, T_2 and T_3 be phylogenetic trees with the same number of internal branches and the same sets of leaves. Then, if:*

1. *For any two bipartitions a and b from different trees: $a \rightarrow b$ implies that $b \rightarrow a$, and*
2. *For any three bipartitions $a \in T_1$, $b \in T_2$ and $c \in T_3$: $a \rightarrow b$ and $b \rightarrow c$ implies that $a \rightarrow c$,*

then, the triangle inequality, $bd(T_1, T_2) \leq bd(T_1, T_3) + bd(T_2, T_3)$, holds.

Proof. On one hand, considering the first statement of Proposition: $bd(T_1,T_2)=(\sum_{\substack{a\in BT_1,\\(b\in BT_2)\to a}} d(a,b)+\sum_{\substack{b\in BT_2,\\(a\in BT_1)\to b}} d(b,a))/2=\sum_{\substack{a\in BT_1,\\(b\in BT_2)\to a}} d(a,b)$, where $(a\in BT_1$ and $(b\in BT_2)\to a)$ means that the sum is taken for all the a's belonging to the bipartition table BT_1 corresponding to the tree T_1 and all the b's associated with these a's. In a similar way:

$bd(T_1,T_3)=(\sum_{\substack{a\in BT_1,\\(c\in BT_3)\to a}} d(a,c)+\sum_{\substack{b\in BT_3,\\(c\in BT_1)\to a}} d(c,a))/2=\sum_{\substack{a\in BT_1,\\(c\in BT_3)\to a}} d(a,c)$, and

$bd(T_2,T_3)=(\sum_{\substack{b\in BT_2,\\(c\in BT_3)\to b}} d(b,c)+\sum_{\substack{c\in BT_3,\\(b\in BT_2)\to c}} d(c,b))/2=\sum_{\substack{b\in BT_2,\\(c\in BT_3)\to b}} d(b,c)$ Consider the following three sums: $\sum_{\substack{a\in BT_1,\\(b\in BT_2)\to a}} d(a,b)$, $\sum_{\substack{a\in BT_1,\\(c\in BT_3)\to a}} d(a,c)$ and $\sum_{\substack{b\in BT_2,\\(c\in BT_3)\to b}} d(b,c)$. Because the Hamming distance d satisfies the triangle inequality, for any term $d(a,b)$ from the first sum we have the term $d(a,c)$ from the second sum and the term $d(b,c)$ from the third sum such that: $d(a,b) \le d(a,c) + d(b,c)$. Because each of the bipartition vectors included in the bipartition tables BT_1, BT_2 and BT_3 appears only once in each of the three sums we conclude that: $bd(T_1, T_2) \le bd(T_1, T_3) + bd(T_2, T_3)$. □

Proposition 2. *The value of the bipartition dissimilarity between two phylogenetic trees on the same sets of n leaves ranges from* 0 *to* $n(n-3)/2$ *if n is even, and from* 0 *to* $(n-1)(n-3)/2$ *if n is odd.*

Proof. For any two binary vectors a and b of size n, the maximum value of the quantity $Min(d(a,b);d(a,\bar{b}))$, where $d(a,b)$ is the Hamming distance between a and b, and $\bar{a}$ and $\bar{b}$ are their complements, is $n/2$ when n is even and $(n-1)/2$ when n is odd. On the other hand, the maximum number of internal branches in a phylogenetic tree (i.e., number of rows of the corresponding bipartition table) with n leaves is $n-3$. Consequently, according to Formula 2, the maximum value of the bipartition dissimilarity between two trees with n leaves is $n(n-3)/2$ if n is even, and $(n-1)(n-3)/2$ if n is odd. □

Theorem 1. *If the newly-formed subtree Sub_{yw} resulting from the HGT (i.e. the subtree rooted by the branch (x,a) in Fig. 2) is present in the gene tree T', and the bipartition vector associated with the branch (x,x_1) in the transformed species tree T_1 (Fig. OA1) is present in the bipartition table of T', then the HGT from (x,y) to (z,w), transforming T into T_1, is a part of* ***a minimum-cost HGT scenario*** *transforming T into T' and satisfying the subtree constraint.*

Proof. The four possible cases leading to the formation of the subtree Sub_{yw} are the following: 1) HGT from (x,y) to (z,w); 2) HGT from (z,w) to (x,y); 3) HGT from (x',x) to (z,z'); 4) HGT from (z,z') to (x',x). When the path (x,z) in T consists of two or more branches, the HGTs corresponding to the cases (3) and (4) will not produce the subtree Sub_{yw}, but bring the vertices x and z closer to each other by reducing the number of branches of the path (x,z). The HGT cases (3) and (4) will induce the bipartition **b**, which will be present in the bipartition table the gene tree T' because of the subtree constraint, such that the leaves of the subtree located to the left of x' and those of the subtree located to the right of z' (Fig. OA1) belong to the same part of it (e.g., they are denoted by 1's in the bipartition table of T'), whereas the leaves of the subtree located below the vertices y and w belong to a different part of it (e.g., they are denoted by 0's in the bipartition table of T'). According to the Theorem condition, the bipartition corresponding to the branch (x,x_1) in the tree T_1 obtained from the initial species tree T after the HGT from (x,y) to (z,w) was carried out, and denoted here $\mathbf{b}_1$, is also present in the bipartition table of T'. This means that the leaves of the subtree located to the left of x' and those of the subtree located below the vertices y and w belong to the same part of it, whereas the leaves of the subtree located to the right of z' belong a different part of it. Obviously, the bipartitions **b** and $\mathbf{b}_1$ are incompatible (i.e., they cannot be present together in the same bipartition table

associated with a phylogenetic tree) meaning that the HGTs from (x',x) to (z,z') and from (z,z') to (x',x) are impossible. Moreover, the HGT from (z,w) to (x,y) is possible only when the path (x,z) in T consists of a single branch (in this case the opposite HGTs from (x,y) to (z,w) and from (z,w) to (x,y) will lead to the same topological transformation of T) because this HGT would induce a bipartition, denoted here $\mathbf{b_2}$, which is incompatible with $\mathbf{b_1}$ if the path (x,z) in T consist of two or more branches. Indeed, in $\mathbf{b_2}$ the leaves of the subtree located to the right of z' and those of the subtree located below the vertices y and w belong to the same part of it, whereas the leaves of the subtree located to the left of x' belong a different part of it. Consequently, the HGT from (x,y) to (z,w) is necessary to transform T into T'. The only exception from this would be the case of the opposite HGT from (z,w) to (x,y) which is possible only if the path (x,z) consists of (or was reduced to) a single branch. In this case the opposite HGTs will lead to the same topological transformation and any of them is a part of a minimum-cost HGT scenario transforming T into T' and satisfying the subtree constraint. □

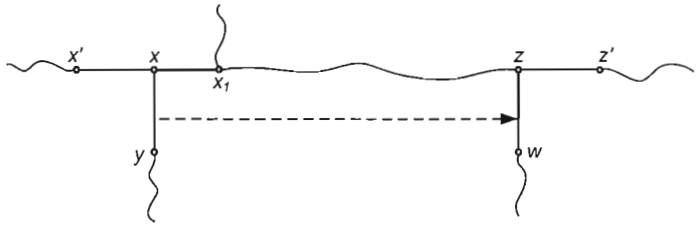


Figure OA1. HGT from the branch (x,y) to the branch (z,w) is a part of a minimum-cost HGT scenario transforming the species tree T into the gene tree T' if the bipartition corresponding to the branch (x,x_1) in the transformed species tree T_1 is present in the bipartition table of T' and the subtree Sub_{yw} (i.e., obtained by the SPR move induced by this HGT, see Fig. 2) is present in T';

Theorem 2. *If the newly-formed subtree $Sub_{y,w}$ resulting from the HGT (i.e. the subtree rooted by the branch (x,a) in Fig. 2) is present in the gene tree T', and all the bipartition vectors associated with the branches of the path (x',z') in the transformed species tree T_1 (Fig. OA2) are present in the bipartition table of T', and the path (x',z') in T_1 consists of at least 3 branches, then the HGT from (x,y) to (z,w), transforming T into T_1, is a part of* ***any minimum-cost HGT scenario*** *transforming T into T' and satisfying the subtree constraint.*

Proof. The bipartition vectors corresponding to the branches (x',x) and (z,z') of the transformed species tree T_1 obtained from T after the HGT from (x,y) to (z,w) are also present in the bipartition table of the species tree T and gene tree T'. Thus, the four possible cases leading to the formation of the subtree $Sub_{y,w}$ are the following: 1) HGT from (x,y) to (z,w); 2) HGT from (z,w) to (x,y); 3) HGT from (x',x) to (z,z'); 4) HGT from (z,z') to (x',x). When the path (x,z) in T consists of two or more branches, the HGTs corresponding to the cases (3) and (4) will not produce the subtree $Sub_{y,w}$, but bring the vertices x and z closer to each other by reducing the number of branches of the path (x,z).

According to the Theorem condition, all the bipartitions of the non-empty path (x,z) in T_1 obtained from the initial species tree T after the HGT from (x,y) to (z,w) are also present in the bipartition table of the gene tree T'. Consequently, the leaves of the subtree located to the left of x' and those of the subtrees located below the vertices y and w (Fig. OA2) belong to a different part (e.g., they are denoted by 1's in the bipartition table of T') of these bipartitions than the leaves of the subtree located to the right of z' (e.g., they are denoted by 0's in the bipartition table of T'). This means that there is no bipartition in T' such that all the leaves located in the subtrees to the left of x' and to the right of z' would belong to one part of it and those from the subtrees located below the vertices y and w, to the other. Thus, the HGT from (x',x) to (z,z'), case (3), as well as the opposite HGT from

(z,z') to (x',x), case (4), will violate the subtree constraint. Obviously, any HGT from the branches (x',x) and (z,z') to the branches of the path (x,z) will also violate the subtree constraint.

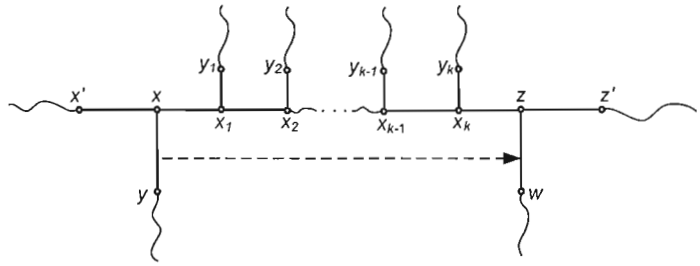


Figure OA2. HGT from the branch (x,y) to the branch (z,w) is a part of any minimum-cost HGT scenario transforming the species tree T into the gene tree T' if all the bipartitions corresponding to the branches of the path (x',z') in the transformed species tree T_1 are present in the bipartition table of T' and the subtree Sub_{yw} (i.e., obtained by the SPR move induced by this HGT, see Fig. 2) is present in the tree T'

Therefore, either the HGT from (x,y) to (z,w) or the opposite HGT from (z,w) to (x,y) is a part of any minimum-cost HGT scenario transforming T into T' and satisfying the subtree constraint. After the HGT from (x,y) to (z,w), all the bipartition vectors corresponding to the branches of a non-empty path (x',z'), in Figure OA2, will be present in the bipartition table of T', and none of them in the case of the opposite HGT from (z,w) to (x,y). As the bipartitions associated with the branches (x_i,x_{i+1}) and (x_{i+1},x_{i+2}), where i = 0,..., k-1, and $x_0 = x'$ and $x_{k+1} = z'$ (Fig. OA2), are present in the bipartition table of T', the bipartition associated with the branch (x_{i+1}, y_{i+1}) is also present in the bipartition table of T'. This means that the subtrees rooted by the branches (x_1,y_1) to (x_k,y_k) can be arranged independently (according to the topology of the gene tree T') if it is not done already, from each other and from the rest of the tree T_1 (i.e., this means that the SPR

operations will be carried out only within these subtrees and inter-subtree SPRs will not be necessary). In the same way, in a minimum-cost scenario the arrangements of the subtrees located to the left of x' and those located to the right of z' (Fig. OA2) should be done independently of the rest of the tree and will take the same minimum number of SPR operations in the case of the HGT from (x,y) to (z,w) and the opposite HGT from (z,w) to (x,y). Consequently, in the case of the opposite HGT from (z,w) to (x,y), the SPR transformation of the tree T_1 into the gene tree T' will take at least one SPR operation more, needed to arrange the branches of the path (x,z), than in the case of the HGT from (x,y) to (z,w). □

Theorem 3. *If the bipartition vectors corresponding to the branches (x,x') and (z,z') of the species tree T (Fig. OA3) are present in the bipartition table of the gene tree T' and the newly-formed subtree, denoted here Sub_{yw}, induced by the HGT (e.g., the subtree rooted by the branch (x,a) in Fig. 2) is present in T', then either the HGT from (x,y) to (z,w) or the opposite HGT from (z,w) to (x,y), transforming the species tree T into T_1, is a part of* ***a minimum-cost HGT scenario*** *transforming T into T' and satisfying subtree constraint.*

Proof. The species tree T (Fig. OA3) can be subdivided into three subtrees by cutting the branches (x,x') and (z,z'). Subtree 1 is rooted by the vertex x' and located to the left of x'; Subtree 2 is rooted by the vertex z' and located to the right of z'; and, Subtree 3 is formed by the subtrees grafted to the path (x,z) and by the branches (x,x') and (z,z'). The fact that the bipartitions associated with (x,x') and (z,z') of the species three T are present in the bipartition table of the gene tree T' means that any minimum-cost scenario transforming T into T' does not include HGTs between the branches of *different Subtrees*, but only those *within* each of them because any HGT between the branches of two different

Subtrees will result in the violation of the subtree constraint (Fig. 2). Any HGT satisfying this constraint preserves all existing identical bipartitions in T and T'.

Consider now Subtree 3. The bipartition vectors corresponding to the branches (x,x') and (z,z') of the species tree T are also present in the bipartition table of the gene tree T'. Assume that the path (x,z) in T consists of a single branch. In this case, the four possible cases leading to the formation of the subtree Sub_{yw} are the following: 1) HGT from (x,y) to (z,w); 2) HGT from (z,w) to (x,y); 3) HGT from (x',x) to (z,z'); 4) HGT from (z,z') to (x',x). Each of these HGTs leads to the same topology of the transformed species tree T_1 and satisfies the subtree constraint, and, consequently, is a part of a minimum-cost scenario transforming T into T'. Thus, when the path (x,z) in T consists of a single branch, the real HGT direction is undetectable.

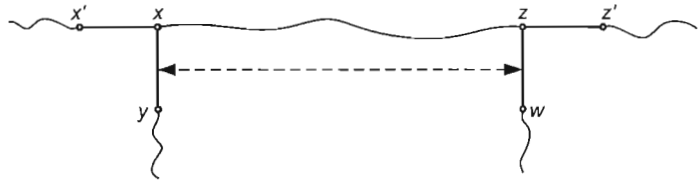


Figure OA3. Either the HGT from (x,y) to (z,w) or the opposite transfer from (z,w) to (x,y) is a part of a minimum-cost HGT scenario transforming T into T' if the bipartitions induced by the branches (x,x') and (z,z') in T are present in the bipartition table of T' and the newly-formed subtree Sub_{yw} resulting from one of these HGTs is present in the tree T'.

Assume now that the path (x,z) in T consists of more than one branch. To form the subtree Sub_{yw} and satisfy the subtree constraint, we can either directly carry out the HGTs (cases 1 and 2) from (x,y) to (z,w) or from (z,w) to (x,y), or regraft by SPR moves all the subtrees of the path (x,z), except those including the branches (x,y) and (z,w), to the

branches (x',x) or (z,z'), and then proceed by the SPR moves (cases 3 and 4) from (x',x) to (z,z'), or from (z,z') to (x',x).

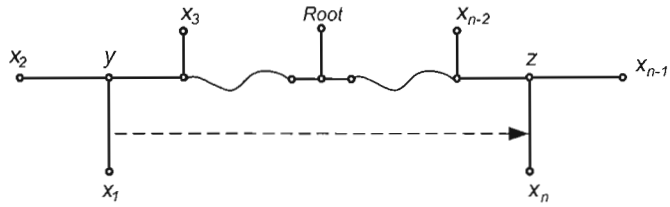
Assume that a minimum-cost scenario S_{min} of the SPR reconciliation of the trees T and T' does not include the HGTs from (x,y) to (z,w) and from (z,w) to (x,y), and proceeds as follows: first, it reduces the path (x,z) to a single branch, and at the last step, merge the vertices x and z to form the subtree Sub_{yw} by the SPR move from (x',x) to (z,z') or from (z,z') to (x',x). It is worth noting that at the last step of the reduction process, the branches (x',x) and (z,z') can be substituted by the other ones before the last SPR move if a HGT between them has taken place beforehand.

We will now show that there is another SPR scenario of the same length including either the HGT from (x,y) to (z,w) or that from (z,w) to (x,y). Without loss of generality assume that in the scenario S_{min} there is a HGT from the branch (x',x) to a subtree grafted to the path (x,z) and induced by the branch denoted here by (x_i,y_i), see Figure OA3, except those induced by (x,y) and (z,w), and that this HGT reduces the path (x,z) to a single branch. In S_{min}, the latter HGT should be followed by another HGT, from (x',x) to (z,z') or from (z,z') to (x',x), initiating the formation of the subtree Sub_{yw}. However, there exists another SPR scenario S, of the same length that S_{min}, which starts by the HGT from (z,w) to (x,y), thus eliminating the vertex x, and brings all the subtrees grafted on the path (x,y), including that induced by the branch (x_i,y_i), one branch closer to the vertex x'. The latter HGT will make the transfer from (x',x) to (x_i,y_i), of the scenario S_{min}, unnecessary. All the other HGTs of S_{min}, that are necessary to arrange the branches grafted to the path (x',z') according to the topology of the gene tree T', will be similar in the scenarios S and S_{min}, thus confirming the optimality of the HGT scenario S. □

Notice: Obviously, in the proofs of Theorems 1, 2 and 3 we assume that the branches (x,y) and (z,w) do not belong to the same lineage. Otherwise, the HGT between them is impossible due to the evolutionary constraints. Also, without loss of generality we assume that T and T' are binary trees.

The RF Metric and SPR Distance

Figure OA4 illustrates a typical situation when the RF metric is unsuitable for finding an optimal scenario of SPR transformations. It shows a HGT in a binary "caterpillar-shaped" tree with n leaves. Here, the species phylogeny T is the tree before the transfer and the gene phylogeny T' is the tree after it. Thus, the SPR distance between T and T' is 1, whereas the RF distance between them equals to its maximum possible value $2n$-6. This example suggests that the RF metric is not a very appropriate measure to approximate the SPR distance. On the other hand, the value of bipartition dissimilarity between T and T' is n-3, whereas its maximum value for the case of two binary trees with n leaves is $n(n-3)/2$ when n is even, and $(n-1)(n-3)/2$ when n is odd (see Proposition 2).



SPR move transforming species tree T into gene tree T'

Figure OA4: The SPR move, representing a HGT, from the branch (x_1, y) to the branch (x_n, z) transforms the species tree T into the gene tree T'. The RF distance between T and T' equals to its maximum value $2n$-6, while the SPR distance between T and T' is 1. In this example, the tree root node is incident to a node of the path (y, z), and the tree leaves are denoted by $x_1 \ldots x_n$.

ONLINE APPENDIX 2

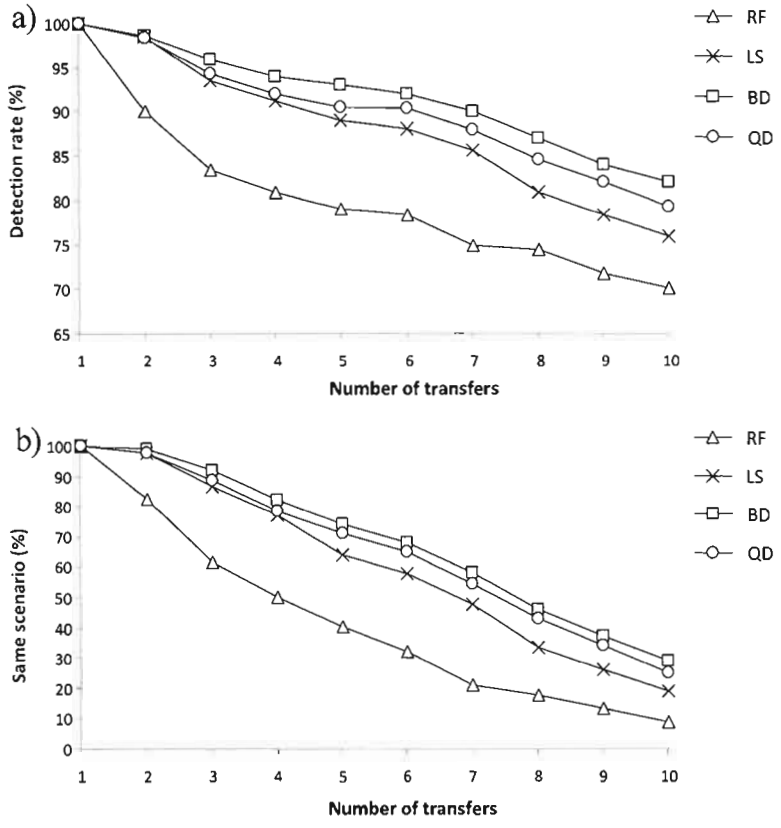


Figure OA5. Percentage of instances the algorithms recover: (a) Correct horizontal gene transfers, and (b) Complete correct HGT scenario, versus the number of HGTs, under the condition of known species and gene trees (i.e., tree-like data). The four compared algorithmic strategies were based on RF, QD, LS and BD. *Each reported value* represents the average result obtained for random trees with 10, 20 .. 100 leaves (1000 replicates were generated for each tree size).

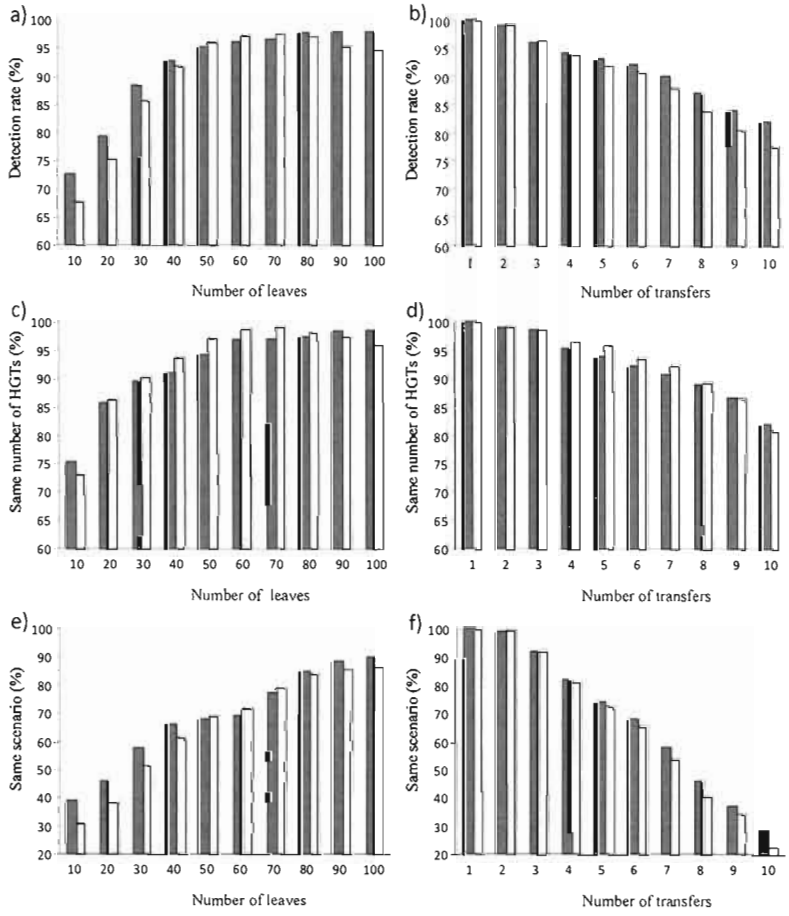


Figure OA6. Percentage of instances when *LatTrans* (white columns) and algorithm based on the bipartition dissimilarity (grey columns) recover: *Correct HGTs* (cases a and b), *Correct total number of HGTs* (cases c and d) and *Complete correct HGT scenario* (cases e and f) depending on the number of tree leaves (cases a, c and e) and number of HGTs (cases b, d and f). *Each reported value* represents the average result obtained for the set of random trees with 1 to 10 HGTs (cases a, c and e) and 10, 20 ... 100 leaves (cases b, d and f); 1000 replicates were generated for each number of HGTs and each tree size.

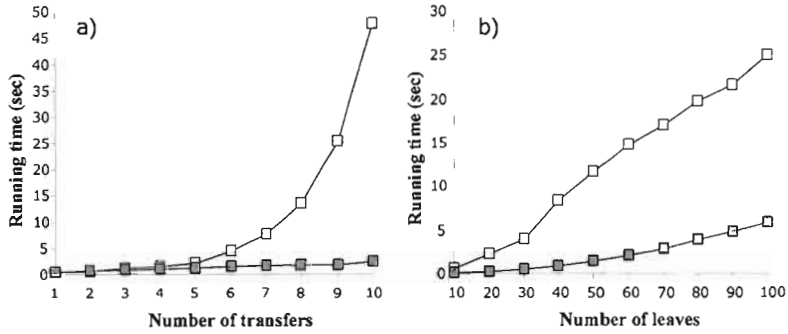


Figure OA7. Running time in seconds for *LatTrans* (white squares) and algorithmic strategy based on the bipartition dissimilarity (grey squares) depending on the: (a) Number of transfers, and (b) Number of tree leaves. *Each reported value* represents the average result obtained for the set of random trees with: (a) 1 to 10 HGTs, and with (b) 10, 20 ... 100 leaves (1000 replicates were generated for each number of HGTs and each tree size).

ONLINE APPENDIX 3

This Appendix includes:

1) An illustration of computing HGT bootstrap support by the *RIATA-HGT* program.

2) The input data for the *rpl12e* and *PheRS Synthetase* examples and the exact output data provided by the *RIATA-HGT* program. Both the text output and solution screenshots are reported in this Appendix.

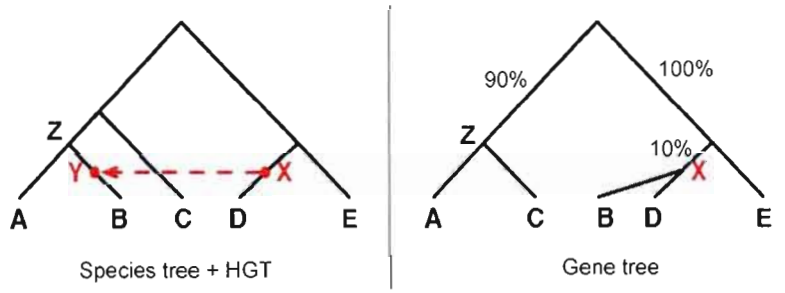


Figure OA8. Computing the bootstrap support of a HGT branch with *RIATA-HGT* The score of the HGT branch $X \rightarrow Y$ added to the species tree is defined as the maximum bootstrap score of all internal branches of the path linking the nodes Z and X in the gene tree. The bootstrap support of the event $X \rightarrow Y$ given by *RIATA-HGT* in this case is 100%. In our method, the bootstrap support of this HGT event would be at most 10%.

RIATA-HGT output for the rpl12e and PheRS Synthetase examples

Input data 1 (Example of the *gene rpl12e*):

(((A.pernix, S.solfataricus), P.aerophilum), (((P.abyssi, P.horikoshii), P.furiosus), ((M.jannaschii, M.thermoaut.), ((T.acidophilum, F.acidarinanus), (((Halobacterium.sp., H.marismortui), M.barkeri), A.fulgidus)))));

(((((P.aerophilum::0.0, S.solfataricus::0.0)::74.0, A.pernix::0.0)::79.0, T.acidophilum::0.0)::79.0, F.acidarinanus::0.0)::100.0, (((P.horikoshii::0.0, P.furiosus::0.0)::61.0, P.abyssi::0.0)::81.0, (((((Halobacterium.sp.::0.0, H.marismortui::0.0)::100.0, M.thermoaut.::0.0)::56.0, M.barkeri::0.0)::56.0, A.fulgidus::0.0)::51.0, M.jannaschii::0.0)::65.0)::100.0);

Notice: Bootstrap scores in the gene tree are indicated after "::". Bootstrap scores of the gene tree branches adjacent to the leaves were set to 0.0 in the Newick string, otherwise *RIATA-HGT* was unable to compute the correct HGT bootstrap support.

Output data 1:

```
species tree:
(((A.pernix,S.solfataricus)I10,P.aerophilum)I11,(((P.abyssi,P.horikoshii)I7,P.furiosus)I8,((M.jann
aschii,M.thermoaut.)I5,((T.acidophilum,F.acidarinanus)I3,(((Halobacterium.sp.,H.marismor
tui)I0,M.barkeri)I1,A.fulgidus)I2)I4)I6)I9)I12;
gene tree:
((F.acidarinanus,(((P.aerophilum,S.solfataricus),A.pernix),T.acidophilum)),(((P.horikoshii,P.furio
sus),P.abyssi),(((((Halobacterium.sp.,H.marismortui)I0,M.thermoaut.),M.barkeri),A.fulgidus)
,M.jannaschii)));
There are 3 component(s), which account(s) for 9 solution(s), each of size 5
--------------------------------------------------------------------------------
Component I12:
Subsolution1.
    I0 -> M.thermoaut. (56.0)
    I11 -> F.acidarinanus (100.0)
    I11 -> T.acidophilum (100.0)
--------------------------------------------------------------------------------
    Component I11:
    Subsolution1.
        I11 -> A.pernix (74.0) [time violation?]
    Subsolution2:
        S.solfataricus -> P.aerophilum (74.0)
    Subsolution3:
        P.aerophilum -> S.solfataricus (74.0)
--------------------------------------------------------------------------------
    Component I8:
    Subsolution1:
        P.horikoshii -> P.furiosus (61.0)
    Subsolution2:
        P.furiosus -> P.horikoshii (61.0)
    Subsolution3:
        I8 -> P.abyssi (61.0) [time violation?]
********************************************************************************
Consensus network for this set of gene trees
(((A.pernix,S.solfataricus)I10,P.aerophilum)I11,(((P.abyssi,P.horikoshii)I7,P.furiosus)I8,((M.jann
aschii,M.thermoaut.)I5,((T.acidophilum,F.acidarinanus)I3,(((Halobacterium.sp.,H.marismortui)I0
,M.barke
ri)I1,A.fulgidus)I2)I4)I6)I9)I12;
P.horikoshii -> P.furiosus
P.furiosus -> P.horikoshii
I8 -> P.abyssi
I11 -> A.pernix
S.solfataricus -> P.aerophilum
P.aerophilum -> S.solfataricus
I0 -> M.thermoaut.
I11 -> F.acidarinanus
I11 -> T.acidophilum
```

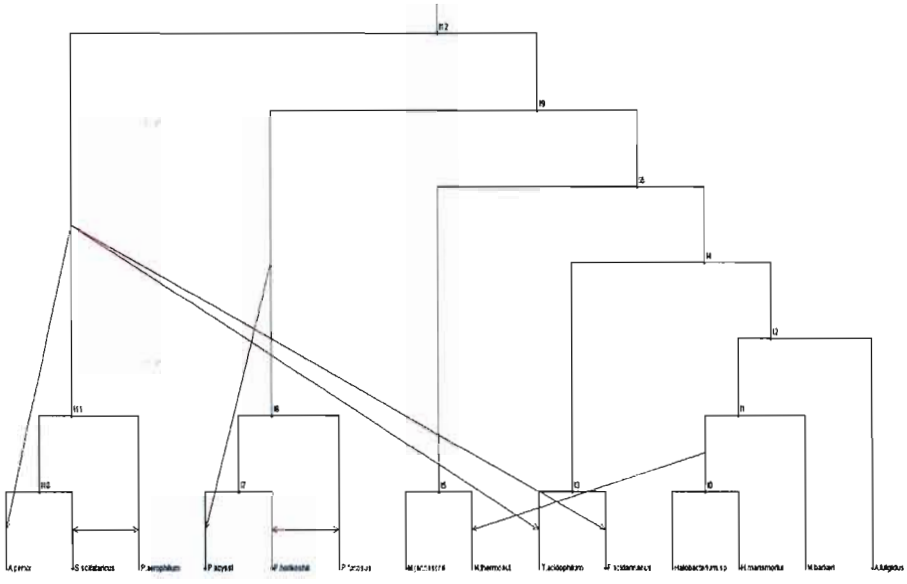


Figure OA9. For the example of the *rpl12e* data *RIATA-HGT* found 9 solutions, each of size 5. Five of these solutions contradict the same lineage constraint (they include HGTs marked by [time violation?] in the program output) and four of them satisfy all plausible evolutionary constraints (e.g., the solution represented in Fig. 6 is among the four eligible solutions). HGT bootstrap scores are indicated between the parentheses in the program output.

Input data 2 (Example of *PheRS Synthetase*):

((((P.hori,M.ther,A.fulg,M.jann),(S.solf,P.aero)),(S.cere,H.sapi)),((B.burg,T.pall),Synech,C.trac,(T.t her,D.radi),(N.gono,H.pilo,(P.aeru,E.coli,H.infl),(R.caps,R.prow)),M tube,T.mari,A.aeol,(P.ging,C. tepi),(C.acet,(B.subt,(E.faec,S.pyog)),(M.pneu,M.geni))));

(((((D.radi::0.0,T.ther::0.0)::100.0,((((N.gono::0.0,P.aeru::0.0)::55.0,((R.prow::0.0,H.pilo::0.0)::67. 0,(H.infl::0.0,E.coli::0.0)::98.0)::34.0),(A.aeol::0.0,Synech::0.0)::19.0):12.0,((C.trac::0.0,P.ging::0.0)::85.0,C.tepi::0.0)::88.0)::8.0)::6.0,((R.caps::0.0,T.mari::0.0)::31.0,(M.tube::0.0,C.acet::0.0)::59.0): .15.0):41.0,((M.pneu::0.0,M.geni::0.0)::100.0,((S.pyog::0.0,E.faec::0.0)::99.0,B.subt::0.0)::90.0)::2 8.0),((((H.sapi::0.0,S.cere::0.0)::100.0,A.fulg::0.0)::24.0,M.ther::0.0)::24.0,(M.jann::0.0,((S.solf::0. 0,P.aero::0.0)::74.0,(P.hori::0.0,(T.pall::0.0,B.burg::0.0)::100.0)::88.0)::85.0)::25.0)::100.0);

Notice: Bootstrap scores in the gene tree are indicated after "::". Bootstrap scores of the gene tree branches adjacent to the leaves were set to 0.0 in the Newick string, otherwise *RIATA-HGT* was unable to compute the correct HGT bootstrap support.

Output data 2:

species tree:
((((P.hori,M.ther,A.fulg,M.jann)I18,(S.solf,P.aero)I17)I19,(S.cere,H.sapi)I16)I20,(((T.ther,D.radi)I13,(N.gono,H.pilo,(R.caps,R.prow)I10,(P.aeru,(E.coli,H.infl)I11)I4)I12,M.tube,T.mari,(Synech,A.aeol)I15,(C.trac,(P.ging,C.tepi)I9)I3,(C.acet,((B.subt,(E.faec,S.pyog)I6)I7,(M.pneu,M.geni)I5)I8)I1)I2,(B.burg,T.pall)I14)I0)I21,
gene tree:
(((((D.radi,T.ther)I13,((((N.gono,P.aeru),((R.prow,H.pilo),(H.infl,E.coli)I11)),(A.aeol,Synech)I15):12.0,((C.trac,P.ging),C.tepi))),((R.caps,T.mari),(M.tube,C.acet))):41.0,((M.pneu,M.geni),((S.pyog,E.faec),B.subt))I8),((((H.sapi,S.cere)I16,A.fulg),M.ther),(M.jann,((S.solf,P.aero)I17,(P.hori,(T.pall,B.burg)I14))))));
There are 3 component(s), which account(s) for 12 solution(s), each of size 14

Component I21.
Subsolution1.
    I17 -> P.hori
    P.hori -> I14 (100.0)
    I16 -> M.ther (25.0)
    I16 -> A.fulg (25.0)
Subsolution2:
    P.hori -> I14 (100.0)
    P.hori -> I17 (85.0)
    I16 -> A.fulg (25.0)
    I16 -> M.ther (25.0)
Subsolution3:
    A.fulg -> I16 (100.0)
    I17 -> M.jann (25.0)
    I17 -> P.hori
    P.hori -> I14 (100.0)
Subsolution4:
    I19 -> M.jann (88.0) [time violation?]
    I16 -> A.fulg (25.0)
    P.hori -> I14 (100.0)
    I16 -> M.ther (25.0)

    Component I2:
    Subsolution1.
        R.prow -> H.pilo (67.0)
        R.caps -> T.mari (31.0)
        I4 -> I3 (85.0)
        I11 -> R.prow (0.0)
        I4 -> I13 (100.0)
        R.caps -> M.tube
        P.aeru -> N.gono (55.0)
        M.tube -> C.acet (59.0)
        I4 -> I15 (19.0)

        Component I3:
        Subsolution1
            P.ging -> C.trac (85.0)

```
        Subsolution2:
            C.trac -> P.ging (85.0)
        Subsolution3:
            I3 -> C.tepi (85.0) [time violation?]
**********************************************************************
Consensus network for this set of gene trees
((((P.hori,M.ther,A.fulg,M.jann)I18,(S.solf,P.aero)I17)I19,(S.cere,H.sapi)I16)I20,(((T.ther,D.radi)
I13,(N.gono,H.pilo,(R.caps,R.prow)I10,(P.aeru,(E.coli,H.infl)I11)I4)I12,M.tube,T.mari,(Synech,
A.aeol)
I15,(C.trac,(P.ging,C.tepi)I9)I3,(C.acet,((B.subt,(E.faec,S.pyog)I6)I7,(M.pneu,M.geni)I5)I8)I1)I2,
(B.burg,T.pall)I14)I0)I21,
P.ging -> C.trac
C.trac -> P.ging
I3 -> C.tepi
R.prow -> H.pilo
R.caps -> T.mari
I4 -> I3
I11 -> R.prow
I4 -> I13
R.caps -> M.tube
P.aeru -> N.gono
M.tube -> C.acet
I4 -> I15
I17 -> P.hori
P.hori -> I14
I16 -> M.ther
I16 -> A.fulg
P.hori -> I17
A.fulg -> I16
I17 -> M.jann
I19 -> M.jann
```

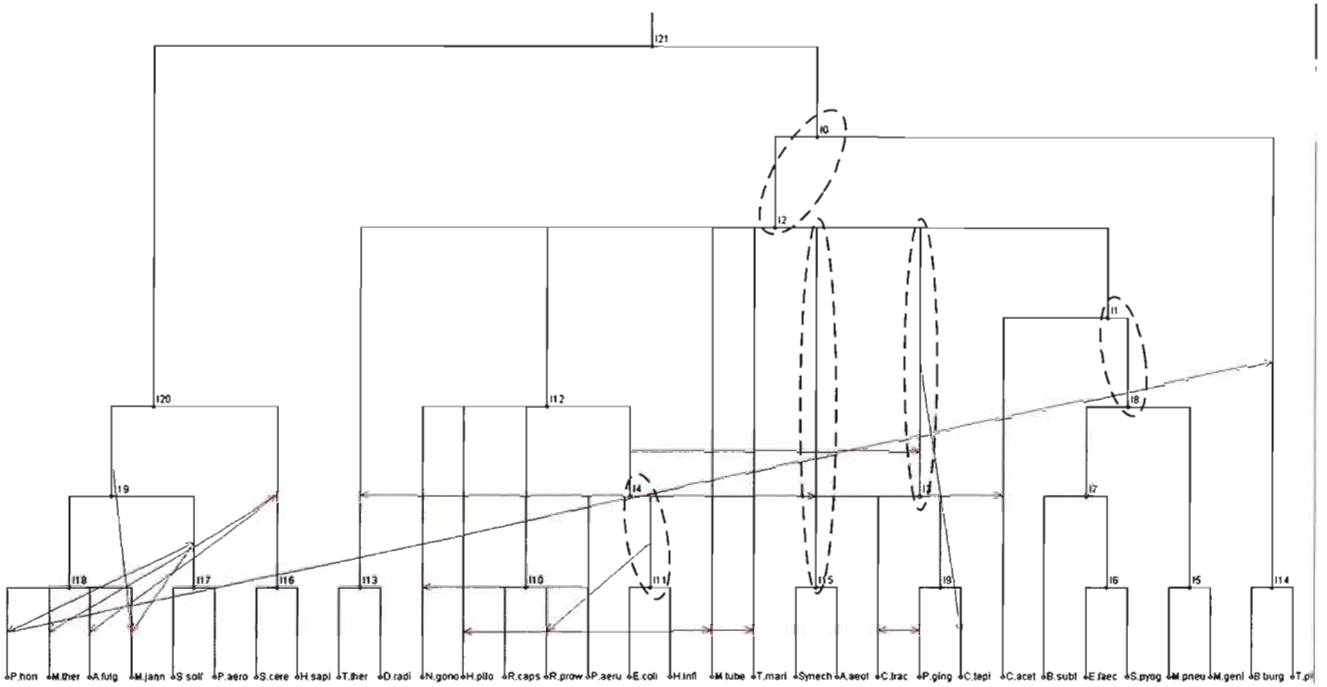


Figure OA10. For the example of the *PheRS Synthetase* data *RIATA-HGT* found 12 solutions, each of size 14. The HGTs contradicting the same lineage constraint are marked by [time violation?] in the program output. Five initial tree transformations indicated by the dashed ellipses were made by *RIATA-HGT* prior to carrying out HGT detection (see the transformed input Newick string of the species tree in the program input). Each of these transformations corresponds to a trivial HGT (i.e., HGTs between the sister taxa from the same multifurcation). Thus, the presented solution actually consists of 19 HGTs, including 14 regular and 5 trivial HGTs. The minimum-cost solution presented in Figure 8, and comprising of 7 regular and 10 trivial HGTs, was not found by *RIATA-HGT*. HGT bootstrap scores are indicated between the parentheses in the program output.

Classification of the Indo-European languages using a phylogenetic network approach

Alix Boc, Anne-Marie Di Scullio and Vladimir Makarenkov

Université du Québec à Montréal
Case postale 8888, succursale Centre-ville Montréal (Québec) H3C 3P8 Canada
boc.alix@courrier.uqam.ca, di_scullio.anne-marie@uqam.ca and
makarenkov.vladimir@uqam.ca

Summary. Discovering the origin of the Indo-European (IE) language family is one of the most intensively studied problems in historical linguistics. Gray and Atkinson [6] inferred a phylogenetic tree (i.e., additive tree or X-tree [2]) of the IE family, using bayesian inference and rate-smoothing algorithms, based on the 87 Indo-European language data set collected by Dyen et al. [5]. When conducting their classification study, Gray and Atkinson assumed that the evolution of languages was strictly divergent and the frequency of borrowing (i.e., horizontal transmission of individual words) was very low. As consequence, their results suggested a predominantly tree-like pattern of the IE language evolution. In our opinion, only a network model can adequately represent the evolution of the IE languages. We propose to apply a method of horizontal gene transfer (HGT) detection [8] to reconstruct phylogenetic network depicting the evolution of the IE language family.

Key words: biolinguistics, historical linguistics, horizontal gene transfer, language classification, phylogenetic network, phylogenetic tree

1 Introduction

A number of curious parallels between the processes of historical linguistics and species evolution have been observed [1, 6, 11]. The evolutionary biologists and historical linguists often look for answering similar questions and face similar problems [1]. Recently, the theory and methodology of the two fields have evolved in remarkably similar ways. A number of important studies have considered the applications of phylogenetic methods to process language data (e.g., [1, 6, 11]). For instance, one of the most intensively studied topics is the evolution of the Indo-European (IE) language family ([4]). Gray and Atkinson [6] inferred a consensus phylogenetic tree of the IE language family using maximal likelihood models of lexical evolution, bayesian inference and rate-smoothing algorithms; the 87 Indo-European language data set collected by Dyen et al. [5] was analyzed in [6]. On the other hand, Rexová et al. [11]

also reconstructed a phylogeny of the IE languages when applying a cladistic methodology to study the same lexicostatistical data set [5]. The results obtained in [11] were very similar to those found in [6]. However, to reconstruct their phylogenies Gray and Atkinson, as well as Rexová et al., were constrained to assume that the evolution of languages was strictly divergent, each language was transmitted as a whole, and the frequency of borrowing (i.e., horizontal transmission of individual words) between languages was low. As consequence, the obtained results suggested a predominantly tree-like pattern of the IE language evolution with little borrowing of individual words.

In our opinion, only a phylogenetic network can adequately represent the evolution of this language family. A network model can incorporate the borrowing and homoplasy (i.e., evolutionary convergence) processes that influenced the evolution of the Indo-European languages. For example, although English is a Germanic language, it has borrowed around 50% of its total lexicon from French and Latin [10].

We propose to apply the methods of horizontal gene transfer (HGT) detection, which are becoming very popular among molecular biologists, in order to reconstruct the evolutionary network of the IE language family. The most frequent *horizontal word transfers*, representing borrowing events, will be added to the phylogenetic tree inferred by Gray and Atkinson (Fig. 1 in [6]) to represent the most important word exchanges which occurred during the evolution of the IE languages. In particular, a HGT detection algorithm ([8]) will be applied to build the evolutionary network of the IE languages.

In this article, we first outline the data in hand and then describe the new features of the HGT detection algorithm used to identify the word borrowing events. In the Results and discussion section, we present the obtained results for the 12 most important groups of the IE languages and report the words borrowing statistics. The most important word exchanges characterizing the evolution of this language family will be brought to light and discussed.

2 Description of the Dyen database

The database developed by Dyen et al. [5] includes the 200 words of the Swadesh list [14]. The Swadesh list is one of several lists of vocabulary with basic meanings, developed by Morris Swadesh in the 1940-50s [14], which is widely used in lexicostatistics (quantitative language relatedness assessment) and glottochronology (language divergence dating). Dyen et al. [5] built a database that provides cognation data among 95 Indo-European speech varieties. For each meaning (e.g., word) in the list of 200 basic meanings (chosen by Swadesh in 1952), the database contains the forms used in the 95 speech varieties and the cognation decisions among the speech varieties made by Isidore Dyen in the 1960s. For each meaning, the forms were examined and cognation judgments were made [5]. The cognation judgments were made only between forms having the same meaning. This is an important aspect of the

lexicostatistical method. The cognation judgments were recorded in classes of forms such that the forms in each class were "cognate" or "doubtfully cognate" with each other. Two forms, in two different speech varieties, were identified as "cognate" if within both of the varieties they had an unbroken history of descent from a common ancestral form. For example, since the English ST word FRUIT and French word FRUIT are known to be related by borrowing, they have been assigned different Cognate Classification Numbers (CCN) in the Dyen database [5]. Forms believed to be related by borrowing or by accidental similarity were thus not treated as cognate. In a small number of cases it was difficult to distinguish cognates from borrowing or accidental similarities; in this case they were treated as "doubtfully cognate" [5]. The cognate content information was used by Gray and Atkinson [6] to reconstruct the evolutionary tree of IE languages. In our study we also subdivided the 200 words of the Swadesh list into two categories : lexical (including nouns and verbs, 138 words in total) and functional (including adjectives, pronouns, conjunctions and determiners, 62 words in total).

3 Materials and methods

In this section, we describe the new features of the HGT detection algorithm [8], applied here in a biolinguistics context, to infer a phylogenetic network of the IE languages family. When applied in a biological context, this algorithm identifies horizontal gene transfers (HGT) of a given gene for a given set of species thus reconciliating the species and gene phylogenetic trees. At each step of the reconciliation process, a HGT event is inferred. In this study, we draw a parallel between the HGT detection and the word borrowing detection processes. In our model, the IE languages tree (Fig. 1 and Fig. 4 in [6]) plays the role of the species tree and the word tree, representing the evolution of a given word (a given translation in all 87 considered languages), plays the role of the gene tree. The algorithmic procedure includes the three main steps, which are as follows:

Step 1. Let L be the rooted tree of 87 IE languages inferred by [6]. Fig. 1 shows a representation of this tree by groups (the group content is reported on the right). We also considered the 200 words of the Swadesh list [14] and their translations into 87 IE languages [5]. For each word of this list, we computed a distance matrix, $\mathbf{D}_i$ (87 x 87), i=1,...,200, between its translations using a normalized Levenshtein distance (Equation 1, [7]).

$$d(i,j) = \frac{Levenshtein\_distance(i,j)}{length(i) + length(j)}. \quad (1)$$

For each such matrix, we inferred the word phylogenetic tree W_i, using the Neighbor Joining method [13]. Fig. 2 shows the Robinson and Foulds (RF) topological distance [12] (normalized by its maximal value of $2n - 6$ for two binary trees with n leaves) between each of the 200 word trees W_i and the

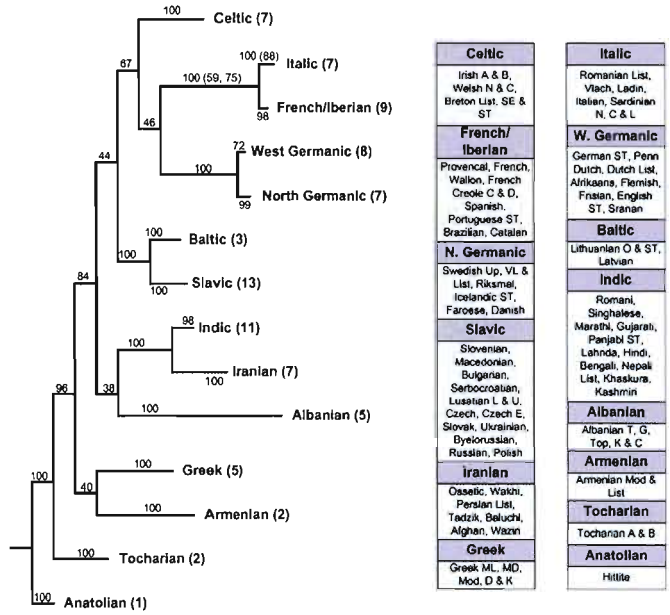


Fig. 1. Gray and Atkinson (Fig. 1 in [6]) IE language evolutionary tree for 14 main language groups. The group content is indicated on the right. The numbers on the tree branches are the tree bootstrap scores; the number of languages for each group is indicated between parentheses.

language tree L. The average value of the normalized RF distance was 82 %. Such a high value suggests an important overall discrepancy between the language tree L and the word trees W_i (i=1,...,200).

Step 2. We applied the HGT detection algorithm ([8]) to infer word borrowings, considering, in turn, the language tree L and each of the 200 word trees W_i. Therefore, 200 different scenarios of tree reconciliation were computed. As the Dyen database [5] did not comprise any translation for the Hittite and Tocharian A and B languages, belonging to the Anatolian and Tocharian groups respectively, these languages were not considered in our analysis.

Step 3. We combined all results from the obtained transfer scenarios to compute the borrowing statistics. Intra-group transfers were ruled out in our computations because of the high risk of accidental similarity among the words from the same language group. First, we assessed the total numbers of transfers (i.e., number of word borrowings) between each pair of groups, and then

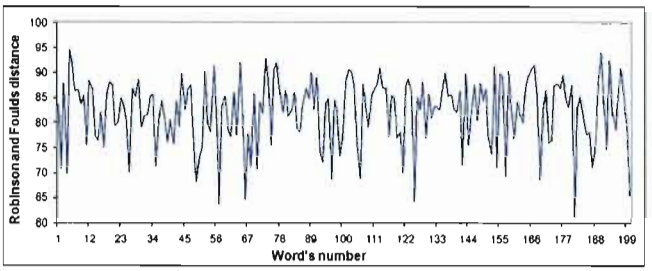


Fig. 2. Normalized Robinson and Foulds topological distance [12] between each of the word trees and the IE language tree in Fig. 1.

the percentages of words affected by these transfers in each group. The 10 most important transfers were mapped into the IE language tree (see Figs. 3, 4 and 5). This computations were carried out, first, for all 200 words and then, separately, for the words from the lexical and functional categories.

4 Results and discussion

Fig. 3 shows the total numbers of borrowed words found for each pair of groups. The 10 most active transfers are highlighted in dark grey. These transfers have been mapped into the IE language tree (Fig. 5a). If the geographical proximity can explain most of the frequent exchanges (e.g., between the West and North Germanic groups), some of them occur between the groups located far away from each other in the tree (e.g., between the Celtic and Indic, or Iranian and Celtic groups).

We can also observe a number of very active exchanges between the cluster combining the Indic and Iranian groups, and that combining the Celtic, Italic, French/Iberian, West/North Germanic and Slavic groups. These results suggest that despite the fact that the Iranian and Celtic groups are located far away from each other in the phylogenetic tree (Fig. 1), there is a strong relationship between them.

Fig. 4 reports the percentages of words of a given group affected by transfers originating from other groups. Similarly to the results reported in Fig. 3, the highest values were found for the neighbor groups. One can also notice that the cluster combining the Indic and Iranian groups has a sustained influence on the other groups. In the same way, we mapped the 10 most intensive transfers into the IE evolutionary tree (Fig. 5b). Some other high percentages (in light grey) can be explained either by well-known historical migration events (e.g., between the Armenian and Iranian groups) or should be investigated in more detail (e.g., between the Slavic and the Albanian groups). For instance,

	Celtic	Italic	French Iberian	W. Germanic	N. Germanic	Baltic	Slavic	Indic	Iranian	Albanian	Greek	Armenian
Celtic	-	53	82	88	58	16	68	89	83	52	30	37
Italic	54	-	357	29	24	10	45	49	32	33	18	1
French/Iberian	33	261	-	34	17	4	17	46	36	36	1	6
West Germanic	36	28	85	-	305	17	44	54	54	22	29	10
North Germanic	36	19	26	192	-	5	16	25	23	21	2	9
Baltic	29	32	23	26	24	-	90	40	46	19	45	6
Slavic	47	45	67	72	35	59	-	80	72	52	22	10
Indic	60	51	64	83	34	26	94	-	161	39	33	17
Iranian	89	41	86	61	43	21	69	224	-	45	25	44
Albanian	48	41	75	26	14	14	47	54	60	-	10	7
Greek	55	28	18	23	11	31	30	68	46	31	-	4
Armenian	43	7	42	22	12	10	20	74	77	21	6	-

Fig. 3. Total numbers of word borrowing events between each pair of language groups. For instance, 53 words of the Italic group were borrowed from the languages of the Celtic group; 10 highest values are highlighted in dark grey and 12 following highest values in light grey.

	Celtic	Italic	French Iberian	W. Germanic	N. Germanic	Baltic	Slavic	Indic	Iranian	Albanian	Greek	Armenian
Celtic	-	3.98	4.36	5.4	4.26	2.65	2.63	4	5.27	5.57	3.07	9.84
Italic	3.6	-	18.99	1.78	1.76	1.66	1.74	2.2	2.03	3.53	1.84	0.27
French/Iberian	2.2	19.58	-	2.09	1.25	0.66	0.66	2.07	2.28	3.85	0.1	1.6
West Germanic	2.4	2.1	4.52	-	22.38	2.82	1.7	2.43	3.43	2.36	2.97	2.66
North Germanic	2.4	1.43	1.38	11.78	-	0.83	0.62	1.12	1.46	2.25	0.2	2.39
Baltic	1.93	2.4	1.22	1.6	1.76	-	3.48	1.8	2.92	2.03	4.61	1.6
Slavic	3.14	3.38	3.56	4.42	2.57	9.78	-	3.6	4.57	5.57	2.25	2.66
Indic	4	3.83	3.4	5.09	2.49	4.31	3.64	-	10.22	4.18	3.38	4.52
Iranian	5.94	3.08	4.57	3.74	3.15	3.48	2.67	10.07	-	4.82	2.56	11.7
Albanian	3.2	3.08	3.99	1.6	1.03	2.32	1.82	2.43	3.81	-	1.02	1.86
Greek	3.67	2.1	0.96	1.41	0.81	5.14	1.16	3.06	2.92	3.32	-	1.06
Armenian	2.87	0.53	2.23	1.35	0.88	1.66	0.77	3.33	4.89	2.25	0.61	-

Fig. 4. Percentages of words affected by borrowing from other groups. For instance, 3.98% of the words of the Italic group have the Celtic origin. The same color notations as in Fig. 3, were adopted here.

Armenian borrowed so many words from the Iranian languages that it was at first considered a part of the Indo-Iranian languages, and was not recognized as an independent group of the Indo-European languages for many decades [15] (see the value of 11.7% for the transfers form Iranian to Armenian in Fig. 4). On the other hand, Baltic languages are extremely well preserved, retaining archaic features similar to ancient Latin and Greek. Similarities of the Baltic languages to ancient Greek (see the value of 5.14% for Greek to Baltic in Fig. 4) and Sanskrit (see value of 4.31% for Indic to Baltic in Fig. 4) were noted long ago by Franz Bopp, the founder of comparative linguistic [3]. Overally, 37% of the considered words were affected by borrowing from other language groups. The analogous results were obtained for the words of the lexical category (36.9%) and functional category (37.1%).

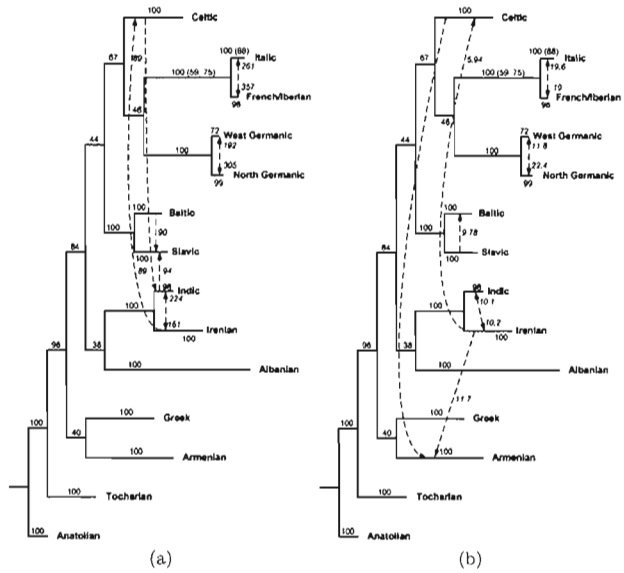


Fig. 5. Ten most frequent word exchanges between the IE language groups in terms of (a) total numbers of transferred words, and (b) percentages of affected words by group.

5 Conclusion

In this paper, we reconstructed a phylogenetic network of the Indo-European language family. The obtained network allowed us to represent the word borrowing events that have an important influence on the evolution of the IE languages. We found that 37% of the IE words are affected by borrowing from other IE groups. Very similar results were obtained for the lexical and functional categories. This means that the word borrowing process does not depend on the word category. However, the obtained result should be interpreted with caution because some of the word similarities, even for words belonging to different language groups, can be due to accidental resemblance. We also found that the clusters combining the Indic and Iranian groups, and the Celtic, Italic, French/Iberian, West/North Germanic groups have much closer relationships that it is represented in the traditional IE tree [6]. This may be the evidence of a much closer common ancestry between these two clusters or of an intensive migration of the ancestors of the involved nations. This finding is very appealing and can certainly bring more light to the ex-

isting hypotheses of the IE language evolution, such as the Kurgan expansion and the Anatolian farming hypothesis. In the future, it would be important to carry out a more comprehensive words borrowing analysis based on the 850 words of the Basic English [9]. Basic English is an English-based controlled language created by Charles Kay Ogden [9] (in essence, a simplified subset of English) as an international auxiliary language. Such a new analysis could help find more recent activities of borrowing. It would be also interesting to establish a parallel between each of the determined high word borrowing activities (see Figs. 3 and 4) and the historical events such as wars, migrations, or important commercial trades between related nations.

References

1. Q.D. Atkinson and R.D. Gray. Curious parallels and curious connections: phylogenetic thinking in biology and historical linguistics. *Syst Biol*, 54:513-26, 2005.
2. J.-P. Barthelémy and A. Guénoche. *Trees and Proximity Representations, Wiley, New York*, 1991.
3. F. Bopp. A Comparative Grammar of the Sanskrit, Zend, Greek, Latin, Lithuanian, Gothic, German, and Slavonic Languages. Translated principally by Lieutenant Eastwick. *London: Madden and Malcolm*, 1845-1856, 1867.
4. J. Diamond and P. Bellwood. Farmers and their languages: the first expansions. *Science*, 300:597-603, 2003.
5. I. Dyen, J.B. Kruskal, and P. Black. Comparative IE Database Collected by Isidore Dyen, http://www.ntu.edu.au/education/langs/ielex/IE-RATE1. 1997.
6. R.D. Gray and Q.D. Atkinson. Language-tree divergence times support the Anatolian theory of Indo-European origin. *Nature*, 426:435-439, 2003.
7. V. I. Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady*, 10:707-710, 1966.
8. V. Makarenkov, A. Boc, C.F. Delwiche, A.B. Diallo, and H. Philippe. New efficient algorithm for modeling partial and complete gene transfer scenarios. In V Batagelj, H.-H. Bock, A. Ferligoj, and A. Ziberna, editors, *IFCS 2006, Series: Studies in Classification, Data Analysis, and Knowledge Organization, Springer Verlag*, 341-349, 2006.
9. C. K. Ogden. Basic English: A General Introduction with Rules and Grammar. Publisher: Paul Treber & Co., Ltd. London, 1930.
10. M. Pagel. In Time Depth in Historical Linguistics In C. Renfrew, A. McMahon, and L. Trask, editors *The McDonald Institute for Archaeological Research, Cambridge, UK* 189-207, 2000.
11. K. Rexová, D. Frynta, and J. Zrzavý. Cladistic analysis of languages: indoeuropean classification based on lexicostatistical data. *Cladistics*, 19:120-27, 2003.
12. D.R. Robinson and L.R. Foulds. Comparison of phylogenetic trees. *Math Biosc*, 53:131-147, 1981.
13. N. Saitou and M. Nei. The neighbor-joining method:a new method for reconstructing phylogenetic trees. *Mol Biol Evol*, 4:406-425, 1987.
14. M. Swadesh. Lexico-statistical dating of prehistoric ethnic contacts: With special reference to north american indians and eskimos. In Proceedings of the *American Philosophical Society*, 96:452-463, 1952.
15. J. Waterman. A history of the German language. *U of Washington Press*, 1976.

Makarenkov et al. BMC Evolutionary Biology 2010, 10:250
http://www.biomedcentral.com/1471-2148/10/250

BMC
Evolutionary Biology

METHODOLOGY ARTICLE Open Access

Weighted bootstrapping: a correction method for assessing the robustness of phylogenetic trees

Vladimir Makarenkov[1*†], Alix Boc[1†], Jingxin Xie[1†], Pedro Peres-Neto[2], François-Joseph Lapointe[3], Pierre Legendre[3]

Abstract

Background: Non-parametric bootstrapping is a widely-used statistical procedure for assessing confidence of model parameters based on the empirical distribution of the observed data [1] and, as such, it has become a common method for assessing tree confidence in phylogenetics [2]. Traditional non-parametric bootstrapping does not weigh each tree inferred from resampled (i.e., pseudo-replicated) sequences. Hence, the *quality* of these trees is not taken into account when computing bootstrap scores associated with the clades of the original phylogeny. As a consequence, traditionally, the trees with different bootstrap support or those providing a different fit to the corresponding pseudo-replicated sequences (the fit quality can be expressed through the LS, ML or parsimony score) contribute in the same way to the computation of the bootstrap support of the original phylogeny.

Results: In this article, we discuss the idea of applying weighted bootstrapping to phylogenetic reconstruction by weighting each phylogeny inferred from resampled sequences. Tree weights can be based either on the least-squares (LS) tree estimate or on the average secondary bootstrap score (SBS) associated with each resampled tree. *Secondary bootstrapping* consists of the estimation of bootstrap scores of the trees inferred from resampled data. The LS and SBS-based bootstrapping procedures were designed to take into account the quality of each "pseudo-replicated" phylogeny in the final tree estimation. A simulation study was carried out to evaluate the performances of the five weighting strategies which are as follows: LS and SBS-based bootstrapping, LS and SBS-based bootstrapping with data normalization and the traditional unweighted bootstrapping.

Conclusions: The simulations conducted with two real data sets and the five weighting strategies suggest that the SBS-based bootstrapping with the data normalization usually exhibits larger bootstrap scores and a higher robustness compared to the four other competing strategies, including the traditional bootstrapping. The high robustness of the normalized SBS could be particularly useful in situations where observed sequences have been affected by noise or have undergone massive insertion or deletion events. The results provided by the four other strategies were very similar regardless the noise level, thus also demonstrating the stability of the traditional bootstrapping method.

Background

In statistics, bootstrapping is a general purpose parameter estimation approach falling within a broader class of resampling methods [1]. Bootstrapping allows one to assess whether the data distribution has been influenced by stochastic effects. Non-parametric bootstrapping proceeds by generating pseudo-replicates of the observed data. Each of the pseudo-replicated data sets is obtained by random sampling with replacement from the original data set. On the other hand, parametric bootstrapping involves sampling from a fitted parametric model, obtained by substituting the maximum likelihood estimator for the unknown population parameter.

Non-parametric bootstrapping is the most commonly used robustness estimation method in phylogenetics [2,3]. It is applied to evaluate the reliability of a phylogenetic tree by examining how often a particular clade, or the corresponding branch, in the tree appears when the original nucleotides or amino acids are resampled. The tree inferring method used to reconstruct the phylogeny from the original data should be carried out to infer the

* Correspondence: makarenkov.vladimir@uqam.ca
† Contributed equally
[1]Département d'informatique, Université du Québec à Montréal, C.P. 8888 succ. Centre-Ville, Montreal (QC) H3C 3P8 Canada
Full list of author information is available at the end of the article

phylogenies from the resampled data. The frequency with which a given branch is found represents its bootstrap support (i.e., bootstrap score).

Different parametric bootstrapping procedures related to phylogenetic analysis were proposed by Huelsenbeck et al. [4], Swofford et al. [5] and Goldman et al. [6]. Parametric bootstrapping can be carried out when we assume an explicit model of sequence evolution. In this case, the original data are used to estimate the stochastic evolutionary parameters, which may include the site-specific rates of evolution, the distribution from which the rates of evolution are drawn or the substitution probabilities on each branch, characterizing the original data set.

In spite of concerns, controversy and confusion over the interpretation of bootstrap scores [7-10], bootstrap analysis has been playing a prominent role in many phylogenetic studies and will likely remain a key method for assessing branch support of phylogenetic trees [11]. It is often assumed, for instance, that the bootstrap support of a branch represents the probability that this branch is correct. However, this point of view is over-simplified [12]. For example, in the case of the famous *Felsenstein zone quartet tree* the maximum parsimony and UPGMA methods converge to the wrong tree as the sequence length increases, and thus both assign very high bootstrap scores to the clades of the wrong phylogeny [13,14]. The best way to interpret the bootstrap support of a given clade is to consider that it indicates the probability that this clade would continue to be found if the same phylogenetic inferring method was applied to pseudo-replicated data having the same empirical distribution as the original data set [12].

In this article we introduce two weighting schemes which can be used to assign weights to each of the trees obtained from pseudo-replicated data. One of them is based on the LS estimate of "pseudo-replicated" trees, whereas the second one proceeds by assessing bootstrap support of those trees (i.e., carries out secondary bootstrapping). These two weighting schemes can be used to correct the standard non-parametric bootstrapping procedure that assign equal weights to each of the phylogenies obtained from the pseudo-replicated sequences. Such a correction will take into account *the quality* of each pseudo-replicated phylogeny. The LS coefficient, as well as the ML function value or the Maximum parsimony score, can be used as an estimate of how close the distance matrix obtained from the pseudo-replicated sequences (or the set of pseudo-replicated sequences, in the case of ML or MP) is to the space of trees. For instance, if it is located far away from this space (i.e., this corresponds to a high value of the LS coefficient) compared to the other trees inferred form pseudo-replicates, then a low weight should be assigned to this tree (and to this pseudo-replicated data set). Alternatively, secondary bootstrapping can be performed to obtain a *robustness estimate* for each of the trees built from pseudo-replicates. Each of the pseudo-replicated multiple sequence alignments (PRA) obtained from the original data can be resampled once again to obtain secondary pseudo-replicated multiple sequence alignments (SPRA) that can be, in turn, used to assess the bootstrap support of the tree inferred from PRA. In this way, an average bootstrap score of internal branches of each pseudo-replicated tree can be used to assign a weight to this tree. Thus, a higher average bootstrap score of a "pseudo-replicated" tree will correspond to a higher weight assigned to this tree.

This article is organized as follows. In the Methods section, we present two weighting schemes, based on the LS and secondary bootstrapping, used to assign weights to "pseudo-replicated" trees. There we also discuss the possibility of normalization of the obtained tree estimates. Then, in the Results section, we present the simulation results for the traditional (unweighted) bootstrapping and four different bootstrapping procedures inducing weights, while considering two real data sets of 12 DNA sequences (*Primate data set* from [15]) and 32 protein sequences (*PheRS sequences* from [16]). In these simulations, we also compare the robustness of the competing bootstrapping procedures by assessing their performances under the condition when different amounts of noise were added to the original data. The Discussion section compares the proposed methods with standard bootstrap correction procedures and explains the rationale of our study. Finally, the Conclusion section summarizes the introduced weighting schemes and presents the ideas for future research.

Methods

Here we discuss four new weighting schemes which can be used in bootstrapping to assign weights to the trees obtained from pseudo-replicated sequences. Specifically, the LS (least-squares) and secondary bootstrap score estimates will be computed for each pseudo-replicated phylogeny. The normalized LS and normalized secondary bootstrap score estimates will be also considered. All these estimates can be used to generate weights of pseudo-replicated trees. A "corrected" bootstrapping procedure based on the obtained weights will be presented.

Let X be a set of n taxa (i.e., objects, species) and A be a multiple sequence alignment obtained for the taxa from X. We assume that each sequence in A has l nucleotides (or amino acids). The model of nucleotide substitution that best fits the data can then be determined and the corresponding data correction applied. A phylogenetic tree T can be inferred by a tree-building algorithm (the Neighbor-Joining [17] algorithm was used in this study to infer phylogenies). The standard non-parametric bootstrap scores can be calculated using the following procedure [2]:

Makarenkov et al. BMC Evolutionary Biology 2010, 10:250
http://www.biomedcentral.com/1471-2148/10/250

(1) l columns of A are randomly chosen with replacements, giving rise to a pseudo-replicated sequence alignment PRA with n rows of l columns. This procedure is repeated N times and a set of pseudo-replicated sequence alignments PRA_1, PRA_2,..., PRA_N is obtained.

(2) Phylogenetic trees T_1, T_2,..., and T_N are then reconstructed from the pseudo-replicated alignments PRA_1, PRA_2,..., and PRA_N, by means of the same tree-inferring algorithm that was used to build T.

(3) The topology of the original tree T is then compared to the topologies of the trees built from pseudo-replicates. The bootstrap score of the branch k in T (denoted here as bs_k) is the percentage of time that k is found in the set of trees T_1, T_2,..., and T_N. It is computed as follows:

$$bs_k = \left(\frac{1}{N}\sum_{i=1}^{N}\sigma_i\right)\times 100\%, \text{ where } \sigma_i = \begin{cases} 1, & \text{if } k \in B_i \\ 0, & \text{if } k \notin B_i \end{cases}, \quad (1)$$

where B_i is the set of internal branches of the tree T_i, given by their non-trivial splits or bipartitions.

LS-based bootstrapping

The least-squares (LS) coefficient can be used to estimate how well the given distance matrix **D**, obtained from the multiple sequence alignment A using a specific sequence-to-distance transformation, approximates the patristic distance (i.e., additive distance or tree metric) Δ between the leaves of the phylogenetic tree T obtained from **D** using the selected tree-building algorithm. In this study, the Jukes-Cantor distance [18] for the DNA sequences and Kimura Protein distance [19] for the amino acids were employed. The least-squares coefficient, LS, between **D** and Δ is computed as follows:

$$ls = \sum_i \sum_j (d(i,j) - \delta(i,j))^2, \quad (2)$$

where $d(i,j)$ is the distance between the taxa i and j, and $\delta(i,j)$ is the patristic distance between the leaves labelled by i and j in the phylogenetic tree T.

We propose to use the LS coefficient to assign individual weights to all trees obtained from pseudo-replicated data (Figure 1). Obviously, the smaller the value of the LS coefficient, the better the phylogenetic tree fits the corresponding distance matrix **D**. Instead of using equal weights for all trees obtained from pseudo-replicated data, as the traditional bootstrapping does, the following four-step weighting scheme was adopted in this study:

1. Given the original sequence alignment A, we first computed from it a series of N pseudo-replicated alignments PRA_1, PRA_2,..., and PRA_N, using the traditional bootstrapping strategy. The Jukes-Cantor [18] evolutionary model was then applied to obtain the distance matrices **M**, $\mathbf{PRM}_1$, $\mathbf{PRM}_2$,..., and $\mathbf{PRM}_N$, from A, PRA_1, PRA_2,..., and PRA_N, respectively. Phylogenetic trees T, T_1, T_2,..., and T_N and the corresponding tree metric matrices Δ, Δ_1, Δ_2,..., and Δ_N were calculated from these distance matrices using Neighbor Joining [17].

2. The vector ls = $\{ls_t \mid t = 1, 2,..., N\}$, comprising the LS coefficients for all N trees obtained from pseudo-replicates was then computed:

$$ls_t = \sum_i \sum_j (d_t(i,j) - \delta_t(i,j))^2, \quad (3)$$

where d_t (i, j) and δ_t (i, j) are, respectively, the distance between the taxa i and j in the pseudo-replicated distance matrix $\mathbf{PRM}_t$ and the patristic distance between the leaves labelled by i and j in the tree t inferred from $\mathbf{PRM}_t$ (Figure 1). The maximum likelihood (ML) and maximum parsimony (MP) estimates can be used at this step as an alternative to LS. In the case of maximum parsimony, multiple optimal trees are usually generated for each replicate (note that multiple trees are possible with ML, although in practice they are not typically recovered). The resultant multiple pseudo-replicated trees can be treated in two following alternative ways: First, a consensus tree for these multiple trees can be established (e.g., using an extended majority rule) and then used in the computations in the same way that the unique NJ tree; second, each of the obtained multiple pseudo-replicated trees can directly contribute to the computation of the weighted bootstrap scores, but the resulting weights (Formulas 4-5) of each of those trees should be in turn divided by the cardinality of the set of optimal trees obtained for the considered set of pseudo-replicated sequences.

3. At this step the weights of all trees,$\mathbf{w} = \{w_t \mid t = 1, 2,..., N\}$, obtained from pseudo-replicates were computed by solving the following system of equations:

$$\begin{cases} \sum_{t=1}^{N} w_t = N, \\ w_1 \times ls_1 = w_2 \times ls_2 = \cdots = w_N \times ls_N. \end{cases} \quad (4)$$

The solution of the system (4) is as follows (for any $t = 1,..., N$):

$$w_t = \frac{N}{ls_t \times \left(\sum_{i=1}^{N} \frac{1}{ls_i}\right)}. \quad (5)$$

4. The LS-based bootstrap scores of the internal branches of T, denoted here as **ls_bs** = $\{ls\_bs_k \mid k = 1,$

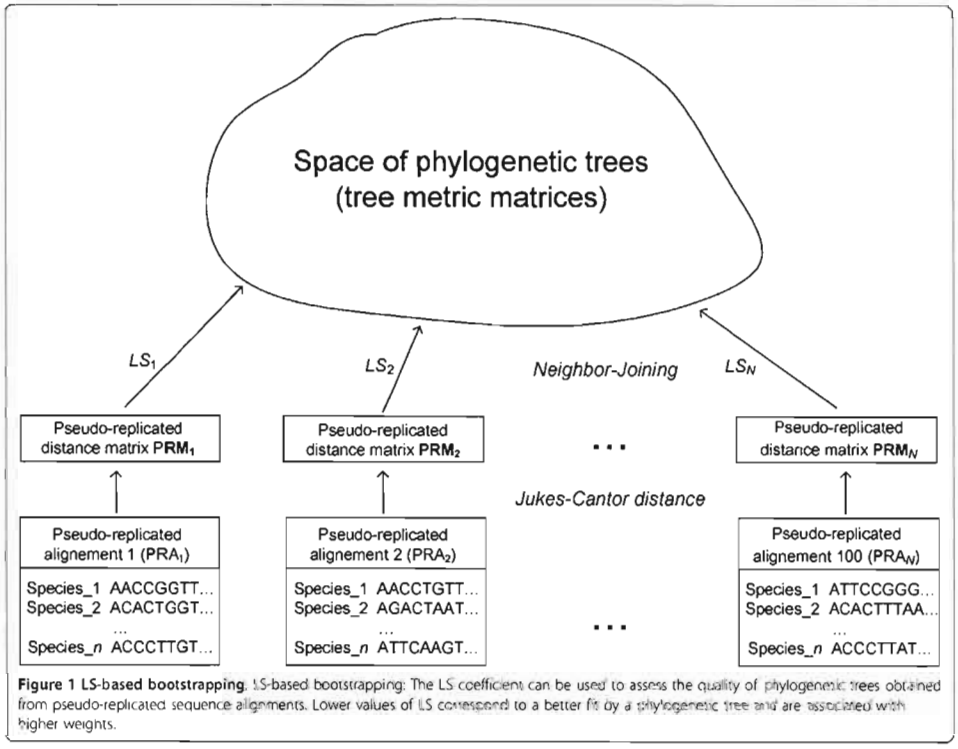


Figure 1 LS-based bootstrapping. LS-based bootstrapping: The LS coefficient can be used to assess the quality of phylogenetic trees obtained from pseudo-replicated sequence alignments. Lower values of LS correspond to a better fit by a phylogenetic tree and are associated with higher weights.

2,..., m}, were then determined. The LS-based bootstrap score of the branch k in T was computed as follows:

$$ls\_bs_k = (\frac{1}{N}\sum_{t=1}^{N} w_t \sigma_t) \times 100\%, \text{ where } \sigma_t = \begin{cases} 1, \text{ if } k \in B_t \\ 0, \text{ if } k \notin B_t \end{cases}, \quad (6)$$

where m is the number of internal branches of the original tree T and B_t is the set of internal branches of the tree t.

Normalized LS-based bootstrapping

Normalized LS-based bootstrap scores can also be computed and used to estimate the robustness of a phylogenetic tree. The normalization of LS, which should in most cases accentuate the difference between the LS coefficients associated with the phylogenetic trees inferred from pseudo-replicated data, was performed in the following way:

$$norm\_ls_t = \frac{ls_t - Min(ls_1, ls_2, \cdots, ls_N)}{Max(ls_1, ls_2, \cdots, ls_N) - Min(ls_1, ls_2, \cdots, ls_N)}, \quad (7)$$

where **norm_ls** = {$norm\_ls_t$ | t = 1, 2,..., N} is the normalized vector of the least-squares coefficients computed after Step 2 (see the four-step weighting procedure described above) and $Min(ls_1, ls_2, ..., ls_N)$ and $Max(ls_1, ls_2, ..., ls_N)$ are, respectively, the minimal and maximal values of the set {ls_1, ls_2,..., ls_N} computed at Step 2. Obviously, all the values of $norm\_ls_t$ (t = 1, 2,..., N) are located in the [0,1] interval. Steps 3 and 4 were then carried out as described above using the normalized LS coefficients, and the weight of the tree t was computed as follows:

$$w_t = \frac{N}{norm\_ls_t \times (\sum_{i=1}^{N} \frac{1}{norm\_ls_t})}, \quad (8)$$

Secondary bootstrapping

Secondary bootstrap scores can be also used to assign weights to phylogenies inferred from pseudo-replicates.

The weight of each phylogeny inferred from (primary) pseudo-replicated sequences can be assessed as the average of bootstrap scores of its internal branches. A pseudo-replicated sequence alignment PRA_i (i = 1,..., N) can be used to create Ns *secondary* pseudo-replicated alignments $SPRA_{i1}$, $SPRA_{i2}$,..., $SPRA_{iNs}$. As in traditional bootstrapping, the columns from PRA_i can be randomly chosen with replacements to create *secondary pseudo-replicates*. Phylogenetic trees T_{i1}, T_{i2},..., and T_{iNs} can then be inferred from the pseudo-replicated alignments $SPRA_{i1}$, $SPRA_{i2}$,..., $SPRA_{iNs}$, and the tree T_i inferred from PRA_i, using the same tree-building algorithm (Figure 2). The topology of T_i can then be compared to the topologies of the trees built from the secondary pseudo-replicates. The bootstrap scores of all internal branches of T_i can be computed, and the average bootstrap score (denoted here as ss_i) characterizing the overall bootstrap support of the tree T_i can be estimated.

When either the ML or MP approach is used, possible multiple optimal pseudo-replicated phylogenies can be treated in two ways: First, the mean of their average bootstrap scores can be taken into account in Formulas 9 and 10 and then their consensus tree in Formula 11; second, each of the obtained optimal MS or ML pseudo-replicated trees can directly contribute to the computation of the weighted bootstrap scores but their resulting weights (Formulas 9-10) should be divided by the cardinality of the set of optimal pseudo-replicated trees.

The weights $\mathbf{w} = \{w_t \mid t = 1, 2,..., N\}$ of all the trees T_i (i = 1,..., N) obtained from primary pseudo-replicates can be computed by solving the following equation system:

$$\begin{cases} \sum_{t=1}^{N} w_t = N, \\ \frac{w_1}{ss_1} = \frac{w_2}{ss_2} = \ldots = \frac{w_N}{ss_N}. \end{cases} \quad (9)$$

The solution of the system (9) is as follows (for any t = 1,..., N):

$$w_t = \frac{N \times ss_t}{(\sum_{i=1}^{N} ss_i)}. \quad (10)$$

Obviously, the bigger the average secondary bootstrap score assigned to a tree, the bigger the tree weight.

The bootstrap scores of the internal branches of T based on secondary bootstrapping, and denoted as $\mathbf{ss\_bs} = \{ss\_bs_k \mid k = 1, 2,..., m\}$, can then be calculated. Thus, the bootstrap score of the branch k in T can be calculated as follows:

$$ss\_bs_k = (\frac{1}{N}\sum_{i=1}^{N} w_i\sigma_i) \times 100\%, \text{ where } \sigma_i = \begin{cases} 1, \text{ if } k \in B_i \\ 0, \text{ if } k \notin B_i \end{cases}, \quad (11)$$

where m is the number of internal branches of the original tree T and B_t is the set of internal branches of the tree t.

Normalized secondary bootstrapping

As in the case of the LS-based bootstrapping, the normalized secondary bootstrap scores can be computed and used to estimate the tree robustness. The normalization, which should emphasize the difference between the average secondary bootstrap scores of phylogenetic trees inferred from primary pseudo-replicates, can be carried out in the following way:

$$norm\_ss_t = \frac{ss_t - Min(ss_1, ss_2, \cdots, ss_N)}{Max(ss_1, ss_2, \cdots, ss_N) - Min(ss_1, ss_2, \cdots, ss_N)}, \quad (12)$$

where $\mathbf{norm\_ss} = \{norm\_ss_t \mid t = 1, 2,..., N\}$ is the normalized vector of the average secondary bootstrap scores of primary pseudo-replicated trees T_1,..., T_N, and Min (ss_1, ss_1, ... ss_N) and $Max(ss_1, ss_1, ... ss_N)$ are, respectively, the minimal and maximal values of the set $\{ss_1,..., ss_N\}$. Then, the weight of the primary pseudo-replicated tree t can be computed as follows:

$$w_t = \frac{N \times norm\_ss_t}{(\sum_{i=1}^{N} norm\_ss_i)}. \quad (13)$$

Results

In this section we apply the four discussed weighting schemes to examine two real data sets consisting, first, of protein-coding mitochondrial DNA sequences for a group of 12 Primate species [15] and, second, of 32 PheRS Synthetase amino acid sequences for a group of 32 organisms, including bacteria, archaea and eukarya [16].

Data description

The first examined data set was originally described by Hayasaka et al. [15]. The latter authors determined nucleotide sequences of homologous 896-base fragments of mitochondrial DNAs (mtDNAs) derived from four species of old-world monkeys, one species of new-world monkeys, two species of prosimians and five species of hominoids. They then reconstructed a phylogenetic tree for this group of 12 Primates. The internal branches of this tree have very high bootstrap support, varying from 85 to 100% (see the Results section). This data set was

Makarenkov et al. BMC Evolutionary Biology 2010, 10:250
http://www.biomedcentral.com/1471-2148/10/250

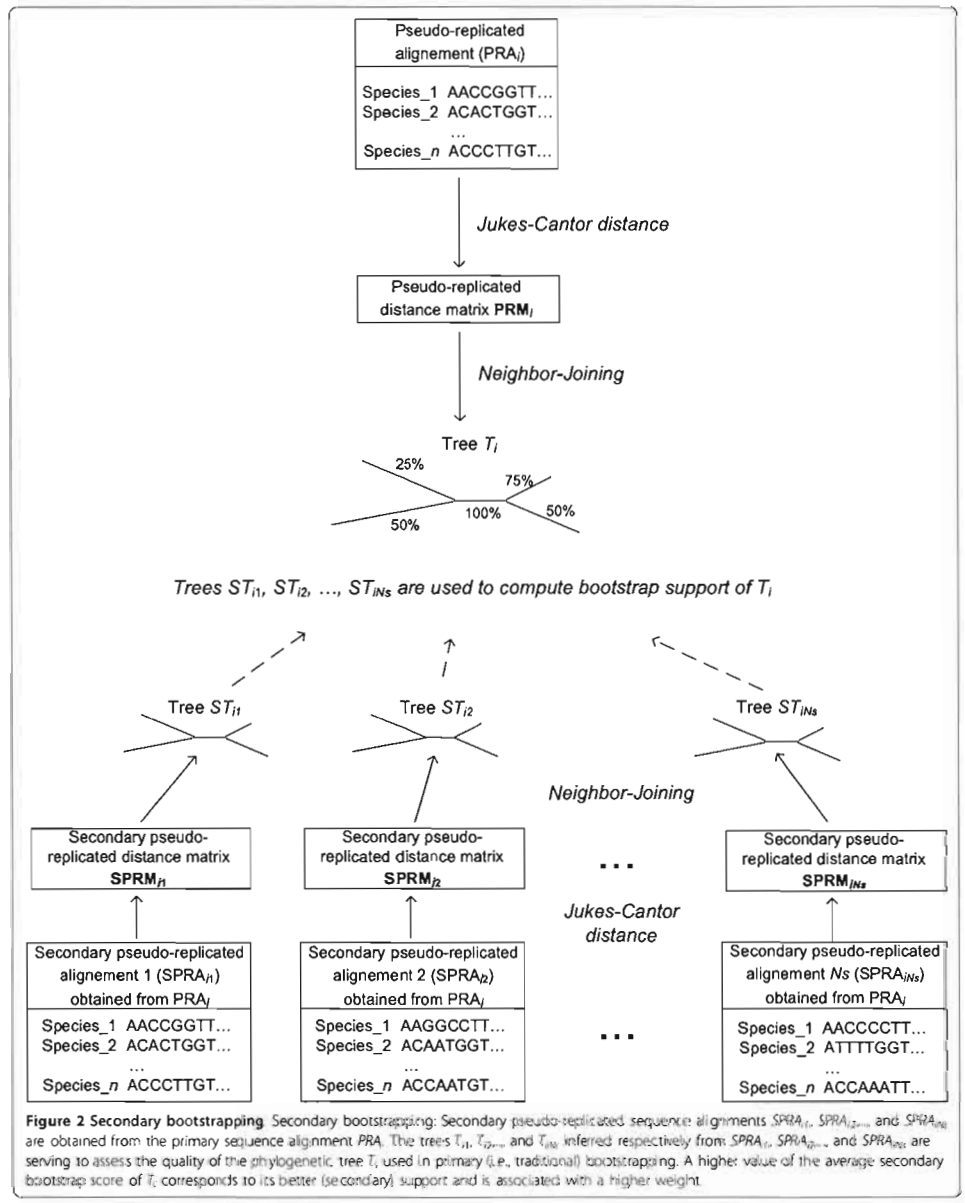


Figure 2 Secondary bootstrapping. Secondary bootstrapping: Secondary pseudo-replicated sequence alignments $SPRA_{l1}$, $SPRA_{l2}$, ..., and $SPRA_{lNs}$ are obtained from the primary sequence alignment PRA_l. The trees T_{i1}, T_{i2}, ..., and T_{iNs} inferred respectively from $SPRA_{l1}$, $SPRA_{l2}$, ..., and $SPRA_{lNs}$ are serving to assess the quality of the phylogenetic tree T_l used in primary (i.e., traditional) bootstrapping. A higher value of the average secondary bootstrap score of T_l corresponds to its better (secondary) support and is associated with a higher weight.

Makarenkov *et al. BMC Evolutionary Biology* 2010, **10**:250
http://www.biomedcentral.com/1471-2148/10/250

also analyzed in a number of evolutionary studies [20-23].

The second considered data set includes 32 PheRS Synthetase sequences with 171 bases for 21 bacteria, 6 archaea and 2 eukarya organisms originally studied by Woese et al. [16]. PheRS is the only class II synthetase in the NUN codon group, and it has no close relatives within that class. For both the α- and β-subunits of PheRS, significant length differences distinguish the bacterial subunits from their archaeal counterparts. Woese et al. [16] found that the AARSs were very informative about the evolutionary process. The analysis of different phylogenetic trees for a number of considered AARSs revealed the following features: The AARSs evolutionary relationships were mostly conform to established species phylogeny; a strong distinction existed between bacterial and archaeal types of AARSs; horizontal transfer of AARS genes between archaea and bacteria was predicted (see also [24]). In fact, PheRS shows classical canonical pattern with the only exception being the spirochetes (i. e., *Borrelia burgdorferi* and *Treponema pallidum*) PheRSs. They are of the archaeal, not the bacterial genre, and are closely related to the clade formed by the archaea *Pyrococcus horikoshii*, *Pyrobaculum aerophilum* and *Sulfolobus solfataricus* (see the Results section and Figure two in [16]). The considered PheRS data set was also studied intensively [24-33].

Distribution of the LS coefficients and average secondary bootstrap scores

First, we examined the distribution of the least-squares (LS) coefficients and secondary bootstrap scores (SBS) for the Primate [15] and PheRS Synthetase [16] data sets presented above (Figures 3 and 4, cases a-b). For both original multiple sequence alignments (MSA), we also created their "noisy" variants by modifying 10% of the nucleotides for the Primate MSA and amino acids for the PheRS MSA (Figures 3 and 4, cases c-d). The noise-affected data were generated in order to investigate how the LS and SBS functions change when the uncertainty is introduced in the data. Figures 3 and 4 show the distribution of LS and SBS for the original (a) and "noisy" (b) MSAs as well as for 100 pseudo-replicated data sets obtained from each of them.

Figures 3a and 4a show that the LS coefficients corresponding to the original MSAs (depicted by encircled diamonds in both figures) are very low (e.g., the lowest LS coefficient in Figure 4a is that of the original MSA). This means that the original MSAs were generally much closer to the space of phylogenetic trees than the pseudo-replicated MSAs obtained from them. After the addition of noise (Figures 3c and 4c) the LS coefficients corresponding to the original and pseudo-replicated MSAs obviously increased, but the difference between them emphasized: The LS coefficient of both original trees (Figures 3c and 4c) became the smallest ones in both cases.

On the other hand, the average SBS corresponding to the original trees (see the encircled triangles in Figures 3b and 4b) were not among the highest ones compared to those of the "pseudo-replicated" trees. This means that the original trees were not necessarily more robust than their pseudo-replicated counterparts. After the addition of noise (Figures 3d and 4d), the robustness of the original and "pseudo-replicated" trees decreased as expected. For the noisy data, the average SBS of the original trees remained only slightly higher than the mean of the average SBS of the "pseudo-replicated" trees.

Simulation study

A simulation study was conducted to evaluate the performances of the four introduced weighting strategies, including the LS and SBS-based (original and normalized) bootstrapping. The traditional bootstrapping scheme, assigning the weights of 1 to all pseudo-replicated trees, was also tested. The simulations were carried out on the Primate [15] and PheRS Synthetase [16] data sets discussed above.

In order to examine the robustness of each weighting strategy, a simulation with "noisy" sequences was performed. A random noise varying from 1 to 10% (with the step of 1%) was added to both original MSAs (for the Primate and PheRS data) to create the variants of "noisy" data. To simulate noisy data in the aligned sequences, we tested two experimental strategies. The first strategy consisted of changing at random a fixed percentage of nucleotides from the observed sequence, whereas the second one consisted of the random elimination or addition of blocks of nucleotides (or amino acids) of different sizes. In this section, we are presenting the combined results (with the 50/50% ratio) for these two strategies detailed below.

Strategy 1. For a given noise percentage (*NR*%), each nucleotide or amino acid of the original data set had the probability of *NR*% to change its state. If the nucleotide or amino acid x was chosen to be affected by noise, it was replaced by a different nucleotide or amino acid. All the other nucleotides or amino acids, different from x, had an equal probability (1/3 for nucleotides and 1/19 for amino acids) to replace x in the MSA. The sequences were not realigned after the addition of noise.

Strategy 2. For a given percentage of noise (*NR*%), the random elimination or addition of blocks of nucleotides (or amino acids) of different sizes (the block sizes were selected randomly and varied from $n*l*NR/2$ to $n*l*NR/10$ nucleotides or amino acids, where n was the number of species and l was the sequence length) was performed. The elimination of blocks of nucleotides or

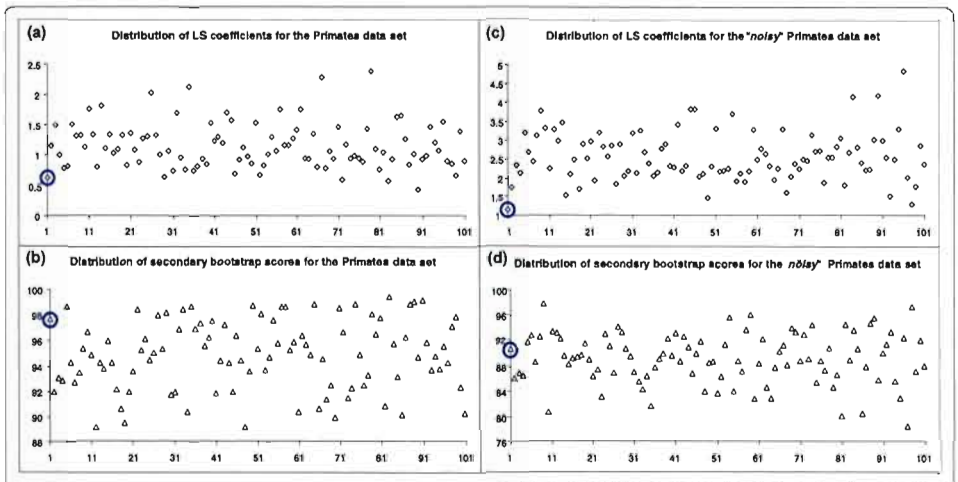


Figure 3 Distribution of the LS coefficients and SBS for the Primate data set. Distribution of the LS coefficients for the original (a) and "noisy" (b) Primate [15] multiple sequence alignments (MSAs) and 100 pseudo-replicated data sets obtained from each of them. Distribution of the average secondary bootstrap scores of the trees corresponding to the original (c) and "noisy" (d) Primate MSAs and 100 pseudo-replicated data sets obtained from each of them. The first (encircled) value corresponds to the original (cases a and b) and noise-affected original (cases c and d) MSAs.

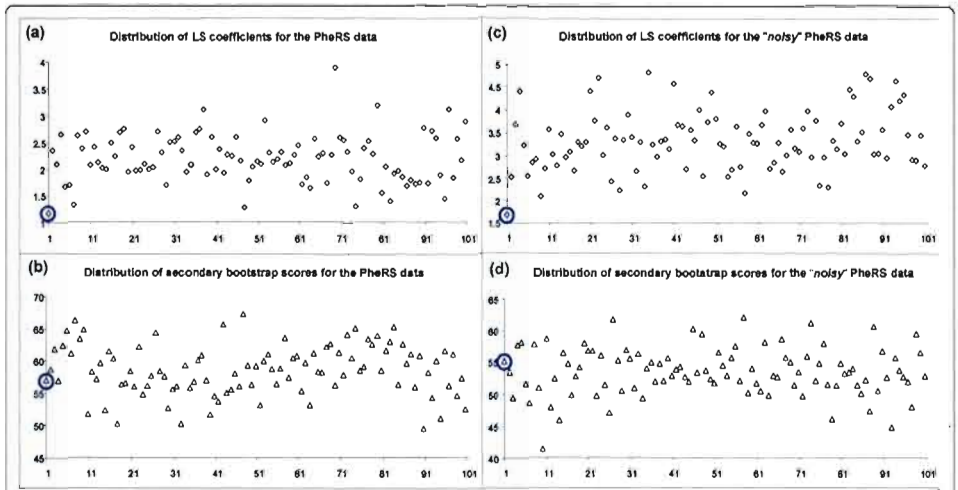


Figure 4 Distribution of the LS coefficients and SBS for the PheRS data set. Distribution of the LS coefficients for the original (a) and "noisy" (b) PheRS [16] multiple sequence alignments (MSAs) and 100 pseudo-replicated data sets obtained from each of them. Distribution of the average secondary bootstrap scores of the trees corresponding to the original (c) and "noisy" (d) PheRS MSAs and 100 pseudo-replicated data sets obtained from each of them. The first (encircled) value corresponds to the original (cases a and b) and noise-affected original (cases c and d) MSAs.

Makarenkov et al. BMC Evolutionary Biology 2010, 10:250
http://www.biomedcentral.com/1471-2148/10/250

amino acids imitates possible deletion events and introduces new gaps in the multiple sequence alignment. The addition of short sequences of nucleotides or amino acids imitates possible insertion events.

The Seqboot program from the PHYLIP package [34] was used to generate multiple resampled versions of the original Primate and PheRS MSA. For each execution, 100 replicates of the original data sets were generated. All the other parameters used were the default Seqboot parameters. The Jukes-Cantor [18], in the case of nucleotides, and Kimura Protein [19], in the case of amino acids, sequence-to-distance transformations followed by the Neighbor-Joining algorithm [17] were carried out to infer phylogenetic trees. The five bootstrapping strategies (4 relying of weights and the traditional one) were tested on such noisy pseudo-replicates. For each of the five strategies, the following measure, denoted here as *least-squares bootstrap deviation* - *ls_bd*, was calculated as follows to assess the strategy robustness:

$$ls\_bd = \sum_{k=1}^{m} (bs_k - bsn_k)^2, \quad (14)$$

where bs_k is the bootstrap score of the internal branch k in the original tree T inferred from the original MSA (i.e., from the original Primate or PheRS data set), bsn_k is the bootstrap score of the internal branch k in the tree T_{noisy} obtained from the original MSA affected by noise, and m is the number of internal branches in the original tree T (note that m was always equal to n-3, where n was the number of species, for both Primate and PheRS phylogenies).

Figures 5 and 6 report, respectively, the Primate [15] and PheRS [16] phylogenies built with Neighbor-Joining. It is worth noting that the Primate phylogenetic tree (Figure 5) perfectly corresponds to that previously obtained by Makarenkov and Legendre [21], whereas the PheRS phylogeny (Figure 6) was different from the tree obtained by Woese et al. [16] and Boc et al. [24], using the ML methods. The most noticeable difference between the presented NJ phylogeny (Figure 6) and the ML trees built by Woese et al. [16] and Boc et al. [24] is that in the tree in Figure 6 the spirochetes (i.e., PheRSs of the bacteria *B. burgdorferi* and *T. pallidum*) are not specifically related to the archaebacterium *P. horikoshii* (these three organisms form a 3-taxon cluster in the trees shown in Figure two in [16] and Figure seven in [24]). The bootstrap scores provided by the five competing bootstrapping strategies (i.e., traditional bootstrap scores, secondary bootstrap scores, LS-based bootstrap scores, normalized secondary bootstrap scores and normalized LS-based bootstrap scores) were calculated for the original and noisy data and depicted in Figure 5 (for the Primate data) and Figure 6 and Table 1 (for the PheRS data). The results presented in Figures 5 and 6, and in Table 1 demonstrate that the normalized secondary bootstrap scores were usually higher than the bootstrap scores yielded by the four other bootstrapping strategies, including the traditional bootstrapping method. This trend was maintained for both original and noisy data. On the other hand, the bootstrap scores provided by the secondary bootstrapping, LS-based bootstrapping, and normalized LS-based bootstrapping were very similar to those obtained with the traditional unweighted bootstrapping. For instance for the original (and, respectively, for the noisy) PheRS data, the standard bootstrap scores were lower than those given by the normalized secondary bootstrap scores strategy in 18 of 29 cases (24 of 29 cases for the noisy data), equal in 9 cases (4 cases for the noisy data) and higher in only 2 cases (1 case for the noisy data). Thus, when a 10%-noise was added to the data, the difference in the bootstrap scores even emphasized. The indicated scores for the original and noisy data, for each of the tested noise percentages, were the averages calculated over 100 repeated calculations (for both primary and secondary bootstrapping).

Moreover, Figures 7 and 8, representing, respectively, the Primate [15] and PheRS [16] data, illustrate the difference in the following parameters between the five bootstrapping strategies: Sum of bootstrap scores of internal branches (Figures 7-8 a-b) and least-squares bootstrap deviation (Figures 7-8 c-d). The latter parameter, computed according to Formula 14, can be viewed as an indicator of the method's robustness. Indeed, the lower the method sensitivity regarding the noise factor, the smaller the least-squares bootstrap deviation. The results in Figures 7-8 are shown depending on the noise percentage (varying from 1 to 10%). When observing the sum of bootstrap scores and the least-squares bootstrap deviation curves, one can notice that the normalized secondary bootstrap scores strategy always provided the highest totals of bootstrap scores of internal branches and the lowest least-squares bootstrap deviations regardless the noise level. For instance for the Primate data set and the normalized secondary bootstrapping, the least-squares bootstrap deviation, *ls_bd*, between the noise-free and noisy bootstrap scores (Formula 14) was equal to 644.01, while for the traditional bootstrapping, the *ls_bd* coefficient was much higher and equal to 786.4. Alternatively, for the PheRS data set and the normalized secondary bootstrapping, the *ls_bd* coefficient was equal to 2279.19, while for the traditional bootstrapping it was also much higher and equal to 2534.58. The additional simulations conducted with larger noise levels (when the noise factor varied from 10 to

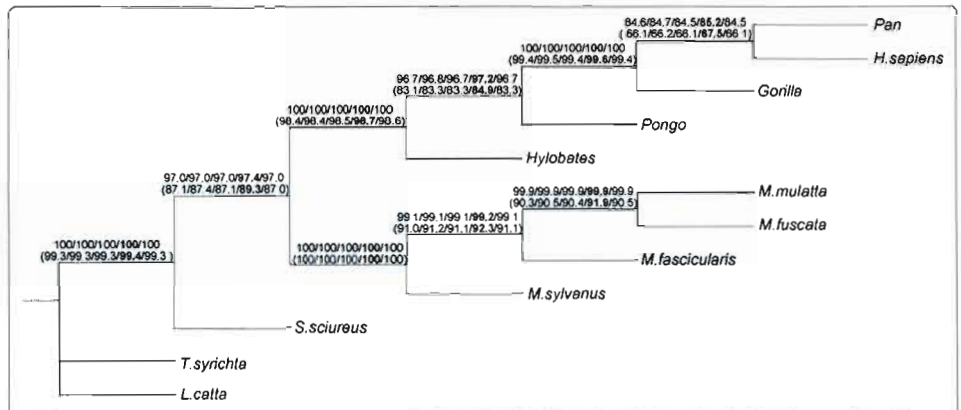


Figure 5 Primate phylogenetic tree with bootstrap scores. The 12-taxon Primate phylogenetic tree inferred with Neighbor-Joining [17]. The nucleotide sequences of 896-base fragments of mitochondrial DNAs [15] were considered and transformed into distances using the Jukes-Cantor transformation [18]. The bootstrap scores provided by the five considered bootstrapping strategies are indicated above internal branches. They are shown for the original and noisy data (the noisy data were obtained after the addition of 10% of noise to the original sequences; they are indicated between parentheses). The bootstrap scores are indicated in the following order: Standard bootstrap scores, secondary bootstrap scores, LS-based bootstrap scores, normalized secondary bootstrap scores (shown in bold) and normalized LS-based bootstrap scores.

35%; these results are not shown) confirmed the observed trend.

Discussion

The controversial study conducted by Hillis and Bull [35] claimed that the traditional bootstrap confidence values used to assess tree accuracy are consistently biased downward. As a response to Hillis and Bull [35], Felsenstein and Kishino [36] argued that the phenomena noticed in [35] are not the result of bootstrap use but rather a result of summarizing the evidence for a given clade using the associated p-values.

Later on, Efron et al. [37] introduced a method for bias correction to estimate more accurate p-values for topological inference through a correction based on first-order p-values. The simplex of possible solutions is partitioned into regions corresponding to different tree topologies [37,38]. Efron's study concluded that the confidence values $\tilde{\alpha}$ obtained using the traditional Felsenstein's bootstrapping are not systematically conservative (i.e., not biased systematically downward) as was stated by Bull and Hillis [35]. Depending on the local configuration of the topological space around the actual tree, the bias may be conservative or liberal. According to Efron's study [37], Felsenstein's method provides a reasonable first approximation to the actual confidence levels of the observed tree clades. One interpretation of non-parametric bootstrapping that is compatible with Bayesian inference was also proposed [[37], page 7090]: "In a Bayesian sense, $\tilde{\alpha}$ can be thought of as reasonable assessments of error". Efron et al. [37] defined another type of non-Bayesian confidence level $\hat{\alpha}$ (which can be estimated by a two-level bootstrap algorithm), such that $\tilde{\alpha}$ and $\hat{\alpha}$ converge at rate $1/\sqrt{n}$, as the sequence length n increases. The methods discussed in [37] and [38] assess the curvature of the solution boundary, which is used in an analytical correction formula to estimate the magnitude of the shifted bootstrap distribution.

In [38], Efron and Tibshirani introduced the "problem of regions". There, one wishes to know which of a discrete set of possibilities applies to a continuous parameter vector. Efron and Tibshirani gave several examples of problem of regions that appear in real applications, including testing significance for model selection and for the number of density peaks. They concluded that, at some point, third-order and higher terms may be necessary to obtain sufficiently accurate confidence estimates [38]. Both of the latter studies used weighting procedures. However, the weights described in [[37] and [38]] are not applied to pseudo-replicated trees, as in our study, but to the first-level bootstrap vectors. The procedure of reweighting the first-order resamples is carried out using a *simple importance sampling scheme* [see the Bootstrap reweighting section in 38 and Equations 4.1-4.14 therein]. According to [38], reweighting the first-order bootstrap samples converts, from a Bayesian point of view, the flat-prior of *a posteriori* probability distribution of the related regions into the appropriate Welch-Peers *a posteriori* probabilities.

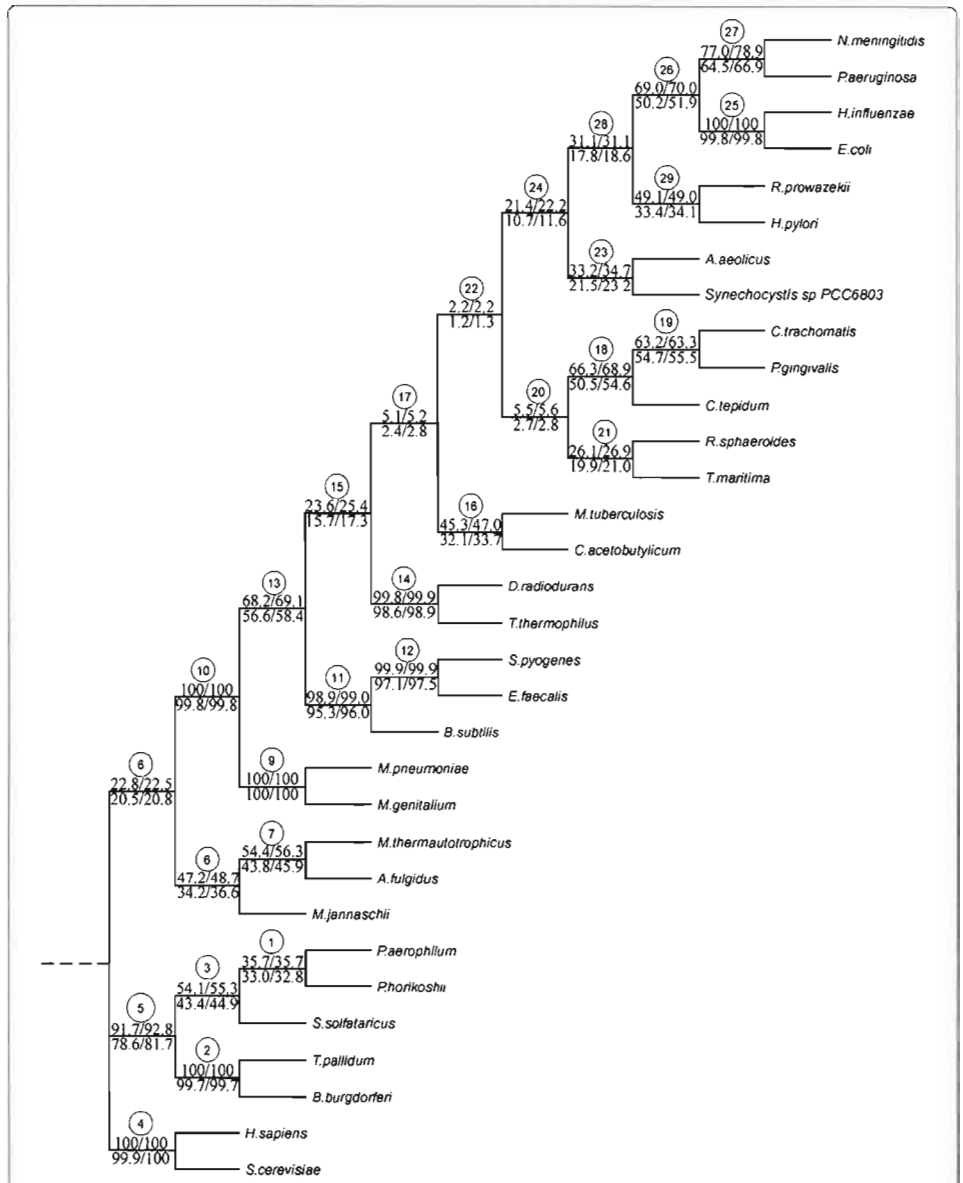


Figure 6 PheRS phylogenetic tree with bootstrap scores. The 32-taxon PheRS Synthetase [16] phylogeny inferred using Neighbor-Joining [17]. The standard bootstrap scores, followed by normalized secondary bootstrap scores, are indicated above the internal branches. They are shown for the original and noisy data (the noisy data were obtained after the addition of 10% of noise to the original sequences; they are indicated under the internal branches). The bootstrap scores for the three other weighting bootstrap strategies are presented in Table 1. Encircled, are the branch numbers as they are reported in Table 1).

Table 1 Bootstrap scores comparison for the PheRS data set

Branch number	*Scores for noise-free data*					*Scores for noisy data (10% of noise)*				
	Std	SB	LS	NSB	NLS	Std	SB	LS	NSB	NLS
1	35.7	35.7	35.4	**35.7**	35.2	33.0	33.0	32.6	**32.8**	32.4
2	100.0	100.0	100.0	**100.0**	100.0	99.7	99.7	99.7	**99.7**	99.7
3	54.1	54.3	53.7	**55.3**	53.4	43.4	43.6	43.0	**44.9**	42.8
4	100.0	100.0	100.0	**100.0**	100.0	99.9	99.9	99.9	**100.0**	99.9
5	91.7	91.9	91.8	**92.8**	91.7	78.6	79.2	78.7	**81.7**	78.7
6	47.2	47.4	47.6	**48.7**	47.9	34.2	34.6	34.6	**36.6**	34.8
7	54.4	54.7	54.4	**56.3**	54.5	43.8	44.2	43.8	**45.9**	43.8
8	22.8	22.7	22.7	**22.5**	22.7	20.5	20.5	20.4	**20.8**	20.3
9	100.0	100.0	100.0	**100.0**	100.0	100.0	100.0	100.0	**100.0**	100.0
10	100.0	100.0	100.0	**100.0**	100.0	99.8	99.8	99.8	**99.8**	99.8
11	98.9	98.9	99.0	**99.0**	99.0	95.3	95.4	95.5	**96.0**	95.6
12	99.9	99.9	99.9	**99.9**	99.9	97.1	97.2	97.0	**97.5**	97.0
13	68.2	68.3	68.8	**69.1**	69.2	56.6	56.9	57.6	**58.4**	58.3
14	99.8	99.8	99.8	**99.9**	99.9	98.6	98.7	98.7	**98.9**	98.7
15	23.6	23.9	23.5	**25.4**	23.5	[illegible]	16.0	15.8	**17.3**	16.0
16	45.3	45.6	45.5	**47.0**	45.6	32.1	32.4	32.4	**33.7**	32.4
17	5.1	5.1	5.0	**5.2**	5.0	2.4	2.5	2.5	**2.8**	2.5
18	66.3	66.8	67.0	**68.9**	67.4	50.5	51.2	51.0	**54.6**	51.4
19	63.2	63.2	63.5	**63.3**	63.7	54.7	54.9	55.0	**55.5**	55.3
20	5.5	5.5	5.6	**5.6**	5.7	2.7	2.7	2.7	**2.8**	2.7
21	26.1	26.2	26.1	**26.9**	26.2	19.9	20.1	20.0	**21.0**	20.0
22	2.2	2.2	2.2	**2.2**	2.3	1.2	1.2	1.2	**1.3**	1.3
23	33.2	33.5	32.9	**34.7**	32.7	21.5	21.8	21.2	**23.2**	21.1
24	21.4	21.5	21.2	**22.2**	21.2	10.7	10.8	10.6	**11.6**	10.7
25	100.0	100.0	100.0	**100.0**	100.0	99.8	99.8	99.7	**99.8**	99.7
26	69.0	69.2	69.4	**70.0**	69.5	50.2	50.5	50.4	**51.9**	50.6
27	77.0	77.3	76.9	**78.9**	77.0	64.5	64.9	64.5	**66.9**	64.6
28	31.1	31.1	31.0	**31.1**	30.9	17.8	17.9	17.8	**18.6**	17.9
29	49.1	49.3	49.3	**49.0**	49.3	33.4	33.5	33.5	**34.1**	33.6

Comparison of bootstrap scores for the five bootstrapping strategies considered in this study. The comparison was made for all 32 internal branches of the phylogenetic trees inferred from the original (i.e., noise-free) and noisy (with 10% of noise added) PheRS sequences. The branch numbers correspond to those indicated in Figure 6. Standard bootstrap scores (Std), secondary bootstrap scores (SB), LS-based bootstrap scores (LS), normalized secondary bootstrap scores (NSB, shown in bold) and normalized LS-based bootstrap scores (NLS) are reported.

Furthermore, the method introduced in this paper is based on the *quality* of pseudo-replicated trees (expressed through the LS and SBS measures) used in the classical Felsenstein's bootstrapping, whereas the Efron method [37], based on an iterative bootstrapping, searches directly for the improvement of the bootstrap scores robustness. It is worth noting that the second-level bootstrap vectors considered in [37] (that are somewhat analogous to the SBS considered in this paper) are generated only for the first-level bootstrap vectors *located on the boundary* of the clade whose robustness is evaluated.

In their recent work, Gullo et al. [39] discussed the usage of different weighting schemes for clustering ensembles, including the problems of consensus tree reconstruction and bootstrap support computation. Clustering ensembles (i.e., consensus clustering or aggregation clustering) have recently emerged as a powerful tool to address traditional clustering issues [39]. Given a data collection, a set of clustering solutions (i.e., ensemble), can be generated by varying the parameter settings. Given a clustering ensemble (in our case, the set of trees obtained from resampled sequences), a major goal is to extract a consensus partition (in our case, the original tree with a robust bootstrap support), taking into account information available from the given set of clustering solutions. Gullo et al. [39] provided the justification for several weighting schemes to discriminate among the clustering solutions, including the one adopted in the present study. Each of these schemes is based on theoretical considerations on ensemble diversity and computes the vector of weights $\mathbf{w} = (w_1,..., w_N)$ in such a way that $w_i \in [0; 1]$, for each $i \in [1,...,N]$, and $\sum_{i=1}^{N} w_i = 1$. The first of those schemes, called *Single*

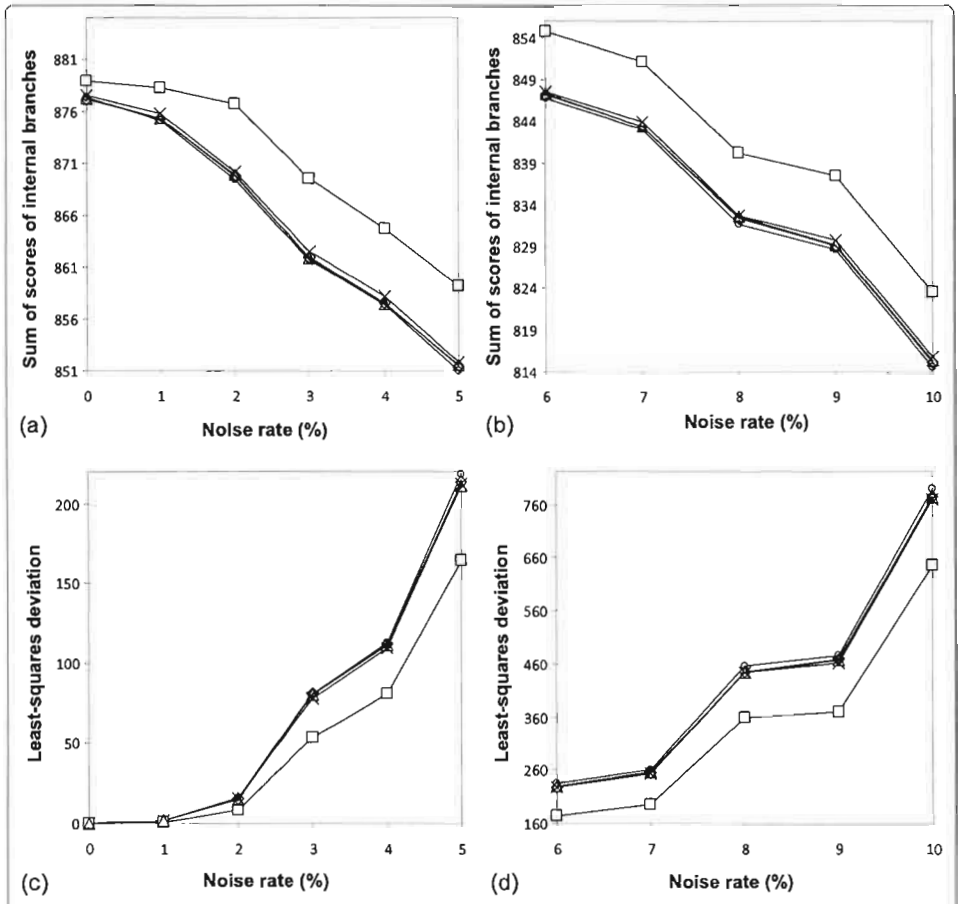


Figure 7 Sum of bootstrap scores and LS deviation for the Primate data set Sum of bootstrap scores of internal branches (cases a and b) and least-squares bootstrap deviation (cases c and d), computed according to Formula 14, for the Primate data set [15] obtained by the five bootstrapping strategies considered in this study. The results are shown with respect to the noise level (the noise percentage varying from 1 to 10% is represented on the *x*-axis). Standard bootstrap scores are depicted by circles, secondary bootstrap scores by crosses, LS-based bootstrap scores by diamonds, normalized secondary bootstrap scores by squares, and normalized LS-based bootstrap scores by triangles.

Weighting (see Formula 4.4 in [39]), presents the most intuitive way to weight each clustering solution (i.e., each pseudo-replicated tree in our case). The key idea consists in computing each individual cluster diversity measure (in our case, such a measure would be the LS coefficient or the average SBS associated with each pseudo-replicated phylogeny) and then in assigning weights that are *proportional* to individual cluster diversities. In fact, Formulas 5 and 10 used in our study to determine the individual weights of pseudo-replicated trees are analogous to Formula 4.4 in [39]. These formulas represent the simplest and the most intuitive way of introducing weights in bootstrap analysis. Most research works focusing on clustering ensembles diversity suggest selecting ensembles according to a maximum diversity criterion [40-42], which states that the higher the ensemble diversity (i.e., the more variation we have in the individual LS coefficients or in the average SBS), the better the accuracy of the consensus partition (i.e., bootstrap scores or consensus tree) extracted

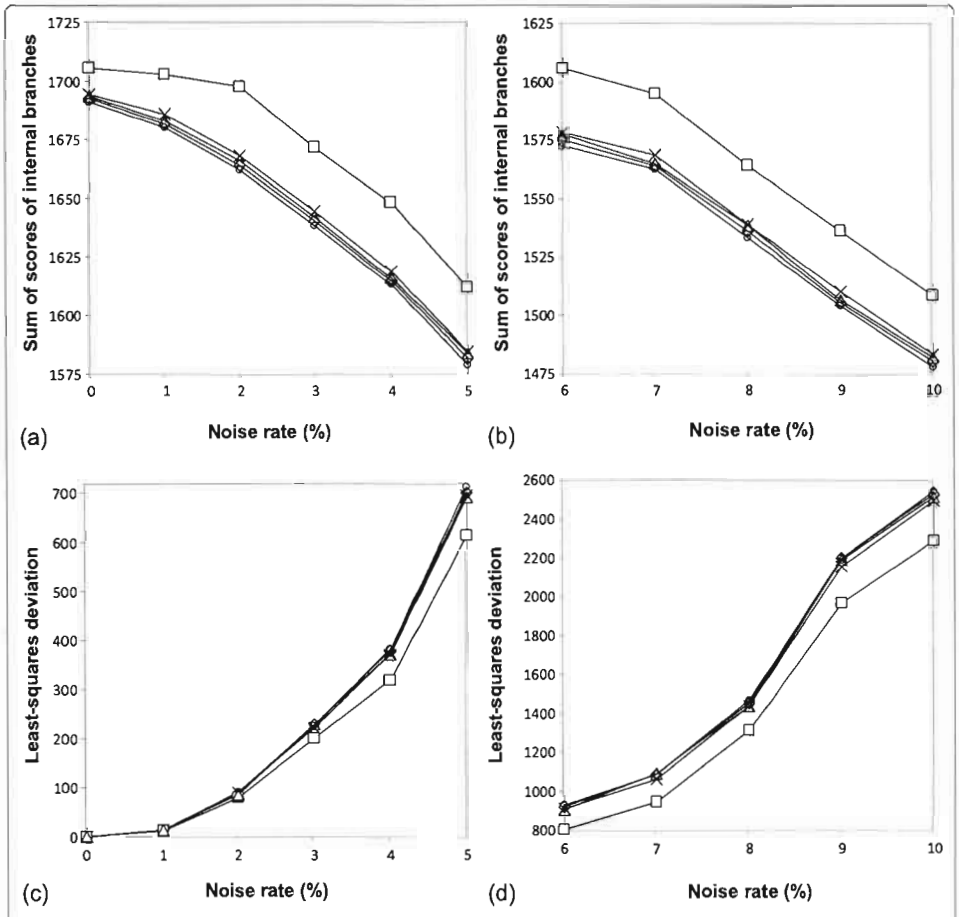


Figure 8 Sum of bootstrap scores and LS deviation for the PheRS data set. Sum of bootstrap scores of internal branches (cases a and b), and least-squares bootstrap deviation (cases c and d), computed according to Formula 14, for the PheRS Synthetase data set [16] obtained by the five bootstrapping strategies considered in this study. The results are shown with respect to the noise level (the noise percentage varying from 1 to 10% is represented on the x-axis). Standard bootstrap scores are depicted by circles, secondary bootstrap scores by crosses, LS-based bootstrap scores by diamonds, normalized secondary bootstrap scores by squares, and normalized LS-based bootstrap scores by triangles.

from the ensemble. Thus, in our study, the weights are computed using a linearly increasing distribution, which defines weights according to a maximum diversity criterion. In the future, it would be also interesting to test the other weighting schemes discussed in [39]. Specifically, a Normal distribution model that computes weights according to a median diversity criterion (see Formula 4.5 in [39]) along with the Group Weighting (see Formulas 4.6-4.9 in [39]) and Dendrogram Weighting (see Formula 4.10 and Algorithm 1 in [39]) models could be tested in the framework of weighted bootstrapping.

Conclusions

The traditional non-parametric bootstrapping is a common method for assessing tree confidence in phylogenetic analysis [2]. It generates and operates pseudo-replicated (i.e., resampled) data sets having the same empirical distribution that the original data set. However, traditional bootstrapping does not take into

account either the "tree-likeness" of phylogenies inferred from pseudo-replicated sequences (i.e., how well these phylogenies fit the corresponding pseudo-replicated sets of sequences) or the bootstrap support of those phylogenies. In this study, we described four weighting strategies allowing one to assign weights to the trees inferred from pseudo-replicates, and thus to do away with one of the limitations of traditional bootstrapping: The assignment of equal weights to all "pseudo-replicated" trees. In our approach, the weights of the trees inferred from pseudo-replicates are assigned according to either the LS estimate of this tree (i.e., how well it fits the pseudo-replicated sequences) or to the average secondary bootstrap scores (SBS) of the tree (i.e., the bootstrap scores associated with the internal branches of "pseudo-replicated" trees). The simulations carried out with two real data sets and five weighting strategies, including the LS and SBS-based bootstrapping, the LS and SBS-based bootstrapping with the data normalization, and the traditional bootstrapping, suggest that the weighted bootstrapping based on the normalized SBS usually exhibits larger bootstrap scores and a higher robustness compared to the traditional bootstrapping and the three other competing methods. The high robustness of the weighting strategy based on the normalized SBS makes this strategy particularly useful in the situations when the considered sequences were affected by noise or underwent insertion or deletion events. Also, when large numbers of replicates (≥100) were considered, the performances of the four other weighting strategies were very similar, thus confirming the stability of the traditional unweighted bootstrapping.

An interesting way for the future investigation would be the study of the proposed weighting schemes in the context of establishing a consensus tree. For instance, the *Consense* program of the PHYLIP package [34] allows the user to introduce weights for each of the input trees. Indeed, the average SBS or LS (original or normalized) estimates of the trees (e.g., of the trees obtained from the pseudo-replicated sequences) could be used to compute the consensus tree (see also [43]). In addition, a new way of computing a consensus tree, which takes into account all individual bootstrap scores of the internal branches of the input trees, could be developed for the weighted supertree methods discussed in [44].

Acknowledgements

The authors thank the Natural Sciences and Engineering Research Council of Canada (NSERC) and Nature and Technologies Research Funds of Quebec (FQRNT) for supporting this research.

Author details

[1]Département d'informatique, Université du Québec à Montréal, C.P. 8888 succ. Centre-Ville, Montreal (QC) H3C 3P8 Canada. [2]Département de sciences biologiques, Université du Québec à Montréal, C.P. 8888 succ. Centre-Ville, Montreal (QC) H3C 3P8 Canada. [3]Département de sciences biologiques, Université de Montréal, C.P. 6128 succ. Centre-Ville, Montréal, Québec, H3C 3J7 Canada.

Authors' contributions

VM, AB and JX designed the methods, implemented them and carried out the simulations. VM, PP-N, F-JL and PL supervised the project and coordinated the development of the methods. All authors read and approved the final manuscript.

Received: 5 March 2010 Accepted: 17 August 2010
Published: 17 August 2010

References

1. Efron B: **Bootstrap Methods: Another Look at the Jackknife.** *Annals Stat* 1979, **7**:1-26.
2. Felsenstein J: **Confidence limits on phylogenies: An approach using the bootstrap.** *Evolution* 1985, **39**:738-791.
3. Penny D, Foulds LR, Hendy MD: **Testing the theory of evolution by comparing phylogenetic trees constructed from 5 different protein sequences.** *Nature* 1982, **297**:197-200.
4. Huelsenbeck JP, Hillis DM, Jones R: Parametric bootstrapping in molecular phylogenetics: Applications and performance. Molecular zoology: Advances, Strategies, and Protocols. In *Symposium held during Annual Meeting of the American Society of Zoologists, St. Louis, Missouri, USA, January 5-8, 1995*. Edited by: Ferraris JD, Palumbi SR. New York, Wiley-Liss Inc; 19-45.
5. Swofford DL, Olsen GJ, Waddell PJ, Hillis DM: **Phylogenetic inference.** In *Molecular Systematics*. Edited by: Hillis DM, Moritz C, Mable BK. Sunderland, Massachusetts, Sinauer Associates; 1996:407-514.
6. Goldman N, Anderson JP, Rodrigo AG: **Likelihood-based tests of topologies in phylogenetics.** *Syst. Biol* 2000, **49**:652-670.
7. Storm CEV, Sonnhammer ELL: **Automated ortholog inference from phylogenetic trees and calculation of orthology reliability.** *Bioinformatics* 2002, **18**:92-99.
8. Burleigh JG, Driskell AC, Sanderson MJ: **Supertree bootstrapping methods for assessing phylogenetic variation among genes in genome-scale data sets.** *Syst. Biol* 2006, **55**:426-440.
9. Seo TK: **Calculating bootstrap probabilities of phylogeny using multilocus sequence data.** *Mol. Biol. Evol* 2008, **25**:960-971.
10. Susko E: **On the Distributions of Bootstrap Support and Posterior Distributions for a Star Tree.** *Syst. Biol* 2008, **57**:602-612.
11. Soltis PS, Soltis DE: **Applying the Bootstrap in Phylogeny Reconstruction.** *Statistical Science* 2003, **18**:256-267.
12. Linder C, Warnow T: **An Overview of Phylogeny Reconstruction.** In *Handbook of Computational Molecular Biology*. Edited by: Aluru S. CRC Press; 2005:.
13. Felsenstein J: **Cases in which parsimony or compatibility methods will be positively misleading.** *Syst. Zool* 1978, **27**:401-410.
14. Felsenstein J: *Inferring Phylogenies* Sunderland, Massachusetts, Sinauer Assoc 2004.
15. Hayasaka K, Gojobori T, Horai S: **Molecular phylogeny and evolution of primate mitochondrial DNA.** *Mol. Biol. Evol* 1988, **5**:626-644.
16. Woese CR, Olsen G, Ibba M, Söll D: **Aminoacyl-tRNA synthetases, the genetic code, and the evolutionary process.** *Microbiol. Mol. Biol. Rev* 2000, **64**:202-236.
17. Saitou N, Nei N: **The neighbor-joining method: a new method for reconstructing phylogenetic trees.** *Mol. Biol. Evol* 1987, **4**:406-425.
18. Jukes TH, Cantor C: Mammalian Protein Metabolism. *Evolution of protein molecules* New York, Academic Press 1969, 21-132.
19. Kimura M: *The neutral theory of molecular evolution* Cambridge University Press, Cambridge 1983.
20. Hayasaka K, Fujii K, Horai S: **Molecular phylogeny of macaques: implications of nucleotide sequences from an 896-base pair region of mitochondrial DNA.** *Mol. Biol Evol* 1996, **13**:1044-1053.
21. Makarenkov V, Legendre P: **Improving the additive tree representation of a given dissimilarity matrix using reticulations.** In *Data analysis, Classification and Related Methods*. Edited by: Kiers HAL, Rasson JP, Groenen PJF, Schader M. Springer; 2000:35-40.

22. Wildman DE, Bergman TJ, al-Aghbari A, Sterner KN. Mitochondrial evidence for the origin of hamadryas baboons. *Mol. Phyl. Evol* 2004, 32:287-296
23. Li D, Fan L, Zeng B, Yin H: The complete mitochondrial genome of Macaca thibetana and a novel nuclear mitochondrial pseudogene. *Gene* 2009, 429:31-36.
24. Boc A, Philippe H, Makarenkov V. Inferring and validating horizontal gene transfer events using bipartition dissimilarity. *Syst. Biol* 2010, 59:195-211
25. Lorsa A, Pan T. Modular construction for function of a ribonucleoprotein enzyme: the catalytic domain of Bacillus subtilis RNase P complexed with B.subtilis RNase P protein. *Nucleic Acids Res* 2001, 29:1892-1897
26. Ambrogelly A, Korencic D, Ibba M: Functional Annotation of Class I Lysyl-tRNA Synthetase Phylogeny Indicates a Limited Role for Gene Transfer. *J. Bacteriol* 2002, 184:4594-4600.
27 Gogarten JP, Doolittle WF, Lawrence JG: Prokaryotic Evolution in Light of Gene Transfer. *Mol. Biol. Evol* 2002, 19:2226-2238.
28 Novichkov PS, Omelchenko MV, Gelfand MS, Mironov AA, Wolf YI, Koonin EV: Genome-Wide Molecular Clock and Horizontal Gene Transfer in Bacterial Evolution. *J. Bacteriol* 2004, 186:6575-6585.
29. Roy H, Ling J, Alfonzo J, Ibba M: Loss of Editing Activity during the Evolution of Mitochondrial Phenylalanyl-tRNA Synthetase. *J. Biol. Chem* 2005, 280:38186-38192.
30. Makarenkov V, Boc A, Delwiche CF, Diallo AB, Philippe H: New efficient algorithm for modeling partial and complete gene transfer scenarios. In *Data Science and Classification*. Edited by: Batagelj V, Ferligoj A, Ziberna A IFCS; 2006:341-349, Series: Studies in Classification, Data Analysis, and Knowledge Organization, Springer Verlag.
31 McAuliffe L, Ellis RJ, Miles K, Ayling RD, Nicholas RAJ. Biofilm formation by mycoplasma species and its role in environmental persistence and survival. *Microbiology* 2006, 152:913-922.
32 Simader H, Hothorn M, Kohler C, Basquin J, Simos G, Suck D: Structural basis of yeast aminoacyl-tRNA synthetase complex formation revealed by crystal structures of two binary sub-complexes. *Nucleic Acids Res* 2006, 34:3968-3979.
33 Brindefalk B, Viklund J, Larsson D, Thollesson M, Andersson SGE: Origin and Evolution of the Mitochondrial Aminoacyl-tRNA Synthetases. *Mol. Biol Evol* 2007, 24:743-756.
34 Felsenstein J: PHYLIP - Phylogeny Inference Package (Version 3.2). *Cladistics* 1989, 5:164-166.
35. Hillis DM, Bull JJ: An Empirical Test of Bootstrapping as a Method for Assessing Confidence in Phylogenetic Analysis. *Syst. Biol* 1993, 42:182-192.
36 Felsenstein J, Kishino H: Is There Something Wrong with the Bootstrap on Phylogenies? A Reply to Hillis and Bull. *Syst. Biol* 1993, 42:193-200
37. Efron B, Halloran E, Holmes S: Bootstrap confidence levels for phylogenetic trees. *PNAS* 1996, 93:13429
38 Efron B, Tibshirani R The problem of regions. *Ann. Statist* 1998, 26:1687-1718.
39 Gullo F, Tagarelli A, Greco S: Diversity-based Weighting Schemes for Clustering Ensembles. *9th SIAM International Conference on Data Mining (SDM '09), Sparks, Nevada, USA, April 30-May 2, 2009* 437-448.
40 Fern X, Brodley C: Random Projections for High Dimensional Data Clustering: A Cluster Ensemble Approach. *In Proc. Int. Conf. on Machine Learning (ICML)*. Edited by: Fawcett T, Mishra N. Washington, D.C., AAAI Press, USA; 2003:186-193.
41 Kuncheva LI, Hadjitodorov ST: Using Diversity in Cluster Ensembles. *Proc. IEEE Int. Conf. on Systems, Man and Cybernetics (SMC), The Hague, The Netherlands* 2004, 2:1214-1219.
42. Domeniconi C, Al-Razgan M: Weighted Cluster Ensembles: Methods and Analysis. *ACM Trans. on Knowledge Discovery from Data (TKDD)* New York, NY, USA 2009, 2:17.1-17.40
43. Lapointe FJ, Cucumel G: The Average Consensus Procedure: Combination of Weighted Trees Containing Identical or Overlapping Sets of Objects. *Syst. Biol* 1997, 46:306-312
44. Ronquist F: Matrix Representation of Trees, Redundancy, and Weighting. *Syst. Biol* 1996, 45:247-253.

doi:10.1186/1471-2148-10-250
Cite this article as: Makarenkov *et al.*: Weighted bootstrapping: a correction method for assessing the robustness of phylogenetic trees. *BMC Evolutionary Biology* 2010 10:250

Nucleic Acids Research Advance Access published September 14, 2011

Nucleic Acids Research, 2011, 1–14
doi:10.1093/nar/gkr735

Towards an accurate identification of mosaic genes and partial horizontal gene transfers

Alix Boc and Vladimir Makarenkov*

Département d'Informatique, Université du Québec à Montréal, C.P.8888, Succursale Centre Ville, Montreal, QC, Canada H3C 3P8

Received January 31, 2011; Revised August 13, 2011; Accepted August 23, 2011

ABSTRACT

Many bacteria and viruses adapt to varying environmental conditions through the acquisition of mosaic genes. A mosaic gene is composed of alternating sequence polymorphisms either belonging to the host original allele or derived from the integrated donor DNA. Often, the integrated sequence contains a selectable genetic marker (e.g. marker allowing for antibiotic resistance). An effective identification of mosaic genes and detection of corresponding partial horizontal gene transfers (HGTs) are among the most important challenges posed by evolutionary biology. We developed a method for detecting partial HGT events and related intragenic recombination giving rise to the formation of mosaic genes. A bootstrap procedure incorporated in our method is used to assess the support of each predicted partial gene transfer. The proposed method can be also applied to confirm or discard complete (i.e. traditional) horizontal gene transfers detected by any HGT inferring method. While working on a full-genome scale, the new method can be used to assess the level of mosaicism in the considered genomes as well as the rates of complete and partial HGT underlying their evolution.

INTRODUCTION

Horizontal gene transfer (HGT) (also called lateral gene transfer) is one of the major mechanisms contributing to microbial genome diversification. HGT is dominant among various groups of genes in prokaryotes (1). The understanding of the key role played by HGT in species evolution has been one of the most fundamental changes in our perception of general aspects of molecular biology in recent years (2,3). HGT can pose several risks to humans, including: cancer triggered by the insertion of transgenic DNA into human cell, antibiotic-resistant genes spreading to pathogenic bacteria, and disease-associated genes spreading and recombining to create new viruses and bacteria (4). Two models of HGT have been considered in the literature (5). First, and the most popular of them, is the traditional model of complete HGT. It assumes that the transferred gene either supplants the orthologous gene of the recipient genome or, when the transferred gene is absent in the recipient genome, is added to it (6). The second model is that of partial gene transfer, implying the formation of 'mosaic' genes. A mosaic gene is an allele acquired through transformation or conjugation (e.g. from a different bacterium) and subsequent integration through intragenic recombination into the original host allele (7,8). The term mosaic stems from the pattern of interspersed blocks of sequences having different evolutionary histories but found combined in the resulting allele subsequent to recombination events. The recombined segments can be derived from other strains of the same species or from other more distant bacterial or viral relatives (7,9). When the incoming DNA is significantly different from the host DNA, mosaic genes can express proteins with novel phenotypes (e.g. in the case when the donor DNA derives from a different species or genus). At the time of HGT event, horizontally transferred genes reflect the base composition of the donor genome. However, over time, these sequences ameliorate to reflect the DNA composition of the host genome because the genes affected by HGT are subject to the same mutational processes that influence all genes in the host genome (10).

There is evidence that mosaic genes are constantly generated in populations of transformable organisms, and probably in all genes (11). Mosaic genes have been also observed in non-transformable bacteria but normally at a lower frequency. Zheng *et al.* (12) reported that mosaic genes account for up to 20% of microbial genomes. For instance, in the naturally competent *Neisseria* species, mosaic alleles have been observed for many genes, comprising those encoding surface antigens, IgA protease,

*To whom correspondence should be addressed. Tel: +1 514 987 3000 (Extn: 3870); Fax: +1 514 987 8477; Email: makarenkov.vladimir@uqam.ca

housekeeping proteins and antibiotic targets (7,10). One of the well-characterized examples of mosaic genes, resulting from partial HGT events, are those that encode the penicillin-resistant binding proteins (PBPs) found in *Streptococcus pneumoniae*. These high molecular weight proteins are the lethal targets of the B-Lactams of penicillin (10,13). Pneumococci, capable of between-species horizontal transfer, undergo, in all likelihood, even more frequent within-species HGT which contributes to the development of mosaic alleles (7).

While many methods have been proposed to address the issue of the identification and validation of complete HGT events (4,6,14–29), only two methods treat the problem of inferring partial HGT and predicting the origins of mosaic genes (30,31). However, neither of the latter two works discusses the problem of robustness of predicted HGT events or includes a Monte Carlo simulation study which is necessary to test the method's performances in different practical situations.

In this article, we describe a new sliding window-based method for predicting partial HGT events and subsequent intragenic recombination. A sliding window approach has been previously used for detecting recombination (32–36), but none of these studies addresses the problem of inferring partial HGT events. The RDP3 program (36) remains, to date, the most comprehensive tool for characterizing recombination events in DNA-sequence alignments. A method for detecting intragenic recombination, called LikeWind, which is also based on a sliding window procedure and on the inference of a phylogenetic tree for each fixed window position, was described in (33). The main advantages of the method we introduce in this article, over LikeWind and the other existing techniques used to detect recombination, are that our method allows one to detect the sources of transferred sequence fragments and assess the robustness of the obtained solution. A Monte Carlo simulation study was carried out to test the ability of the proposed method to recover correct partial HGTs depending on the number of gene transfers and number of species considered (i.e. tree size). In the 'Results' section, the new method is applied to recover partial, and complete, HGT events in the context of the evolution of the genes *rbcL* [data originally considered in Ref. (37)] and *mutU* [data originally considered in Ref. (30)].

MATERIALS AND METHODS

A new method for predicting partial horizontal gene transfer events

In this section we describe the main features of the new method for inferring partial HGTs. The main steps of the method intended to provide an optimal scenario of partial transfers of the given gene for the considered group of species, and thus predict putative intragenic recombination events and identify mosaic sequences, are summarized below. The bootstrap validation will be performed for each predicted partial transfer, and only the transfers with significant bootstrap support will be included in the final solution. A sliding window procedure will be carried out to test different fragments of the given multiple sequence alignment (MSA). A method for detecting complete HGTs will be carried out at each step to reconcile the given species tree and partial gene trees inferred from the sequence fragments located within the sliding window (each time its position is fixed).

Preliminary step. Let X be a set of species, MSA be a given multiple sequence alignment of length l, and $S_{i,j}$ be the MSA fragment under examination located between the sites i and j (including both i and j), where $1 \leq i < j \leq l$. Define the sliding window size w ($w = j - i + 1$) and the progress step size s. Infer the species phylogenetic tree, denoted T. Usually a morphology-based tree or a molecular tree based on a molecule assumed to be refractory to horizontal gene transfer plays the role of the species tree. For instance, 16S rRNA or 23S rRNA genes may also undergo HGT, but they seem to do it at a relatively low rate (38). The tree T must be rooted with respect to the available evolutionary evidence. If no plausible evidence for rooting T exists, the outgroup or midpoint strategies can be used (6). The tree rooting is necessary because it allows us to take into account the evolutionary time-constraints that should be satisfied when inferring HGTs. These time constraints, which include the same lineage HGTs as well as some criss-crossing transfers, are imposed by the necessity for taxa involved in HGT to be contemporaneous (6,18,20). Fix the sliding window size w and the step size s. In our experiments, the window sizes of $l/5$, $l/4$, $l/3$ and $l/2$ sites and the sliding window progress step of 10 sites were used.

Step k. Fix the position of the sliding window in the interval $[i, j]$, where $i = 1 + s(k-1)$ and $j = i + w - 1$; k also corresponds to the window rank (Figure 1). If $i + w - 1 > l$ and $i + w - 1 - l \geq w/2$, then $j = l$, otherwise stop the algorithm here (i.e. short window sizes usually lead to trees with low bootstrap support and hence to doubtful HGTs). Infer a partial gene tree T' characterizing the evolution of the MSA fragment located within the interval $[i, j]$. In this study, the PhyML method (39) was used to reconstruct partial gene trees. Apply an existing HGT detection method to infer an optimal scenario of partial HGTs associated with the interval $[i, j]$. Here we used the HGT-Detection method described in Ref. (6) in the context of complete HGT, but any other HGT inferring method can be carried out instead. The HGT-Detection method was shown to be faster and, in most instances, as accurate as the popular LatTrans (20) and RIATA–HGT (4) methods. Here, the bipartition dissimilarity measure introduced in (6) was used as an optimization criterion. It takes into account the degree of similarity between the topologically closest subtrees in two phylogenetic trees and can be viewed as a refinement of the popular Robinson and Foulds topological distance (40).

In addition, the following procedure for assessing the reliability of obtained partial transfers (i.e. HGT bootstrap support), which takes into account the uncertainty of partial gene trees as well as the number of occurrences of the selected transfers in all minimum-cost HGT

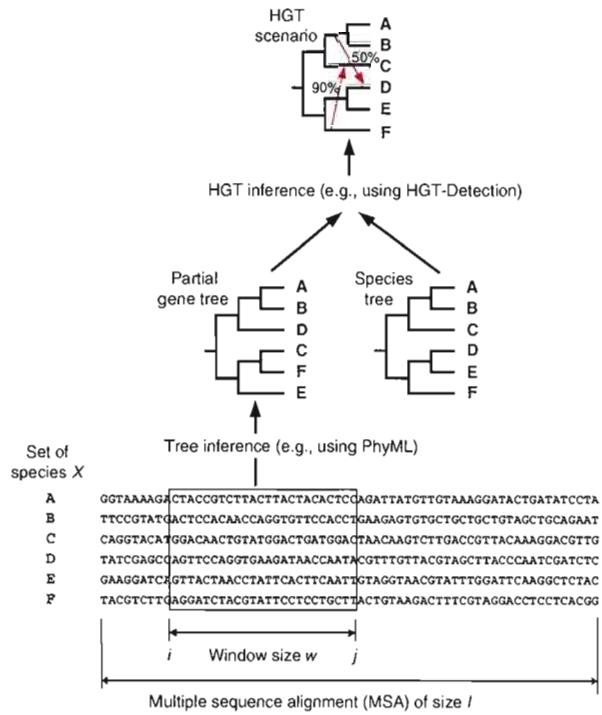


Figure 1. Partial gene tree is inferred using the sequences located within the sliding window of size *w*. The PhyML method (39) was used to reconstruct partial gene trees. The HGT-Detection method (6) was then applied to infer complete HGTs

scenarios found for the given pair of species and partial gene trees, was carried out. In the bootstrap procedure, only the sequence data used to build the partial gene tree T', inferred from the sequences located within the sliding window, were pseudo-replicated. The species tree T was fixed and thus taken as an a priori assumption of the method. We first executed our program with the exhaustive search option providing the list of all minimum-cost HGT scenarios. This option consists of verifying at each step of the algorithm all possible HGTs that satisfy the evolutionary constraints. Once the list of all possible minimum-cost HGT scenarios for the trees T and T' was established, the HGT bootstrap score of each individual partial transfer was computed. Formulas 1 and 2 were used to compute the bootstrap score *HGT_BS* of the partial transfer t:

$$\text{HGT\_BS}(t) = \left(\sum_{1 \le i \le N_{T'}} \left(\sum_{1 \le k \le N_i} \frac{\sigma_{ki}(t)}{N_i} \times 100\,\% \right) \right) / N_{T'} \quad (1)$$

and

$$\sigma_{ki}(t) = \begin{cases} 1, \text{ if the transfer } t \text{ is a part of the minimum-cost scenario } k \text{ for the gene tree } T_i' \\ 0, \text{ if not.} \end{cases} \quad (2)$$

where $N_{T'}$ is the number of partial gene trees (i.e. number of HGT bootstrap replicates) generated from pseudo-replicated sequences and N_i is the number of minimum-cost scenarios obtained when carrying out the algorithm with the species tree T and partial gene tree T_i'. Among the obtained partial HGTs, only the transfers with significant bootstrap scores were retained. Obviously, a short window size produced partial gene trees with much greater variability, and hence lower bootstrap supports for HGT histories.

Final step. Establish a list of predicted partial HGT events. Identify the overlapping intervals giving rise to

the identical partial transfers (i.e. the same donor and recipient and the same direction). Re-execute the HGT detection method for all overlapping intervals (considering their total length in each case) that support the identical partial HGTs. If such partial HGTs are found again for the sequence fragment located within the overlapped intervals, assess their bootstrap support and, depending of the obtained support, include them in the final solution or discard them. If a window located in the middle of a larger interval does not suggest the transfer that is indicated (with a certain significant bootstrap support) on the interval's ends, the entire, larger, interval is tested for the presence of significant HGTs.

The time complexity of the proposed method is as follows:

$$O(r \times (\frac{(l-w/2)}{s} \times (C(Phylo\_Inf)+C(HGT\_Inf)))), \quad (3)$$

where w is the size of the sliding window, s is the sliding window progress step, $C(Phylo\_Inf)$ is the time complexity of the tree inferring method used to infer phylogenies from sequence fragments located within the sliding window, $C(HGT\_Inf)$ is the time complexity of the complete HGT detection method used to infer HGTs for the given species tree and partial species trees inferred from sequence fragments located within the sliding window, r is the number of replicates in bootstrapping.

Given that the time complexity of PhyML (39) is $O(pnw)$, where p represents the number of refinement steps being performed, and the time complexity of HGT-Detection (6) is $O(\tau \times n^4)$, the exact time complexity of our implementation is as follows:

$$O(r \times (\frac{(l-w)}{s} \times (pnw+\tau \times n^4))), \quad (4)$$

where n is the number of species, and τ is the average number of transfers found for a sequence fragment located within the sliding window of size w.

For instance, the running time of the algorithm for the numerical example considered in the Results section and involving an MSA of 30 *mutU* DNA sequences of length 384 sites, three different window sizes: 100, 150 and 200 sites, the advancement step of 10 sites and 100 replicates in HGT and PhyML bootstrapping, was 4 min and 33 s when executed on a PC computer equipped with the Intel Core i7-2635QM (2.0 GHz) processor and 4 Gb of RAM.

RESULTS

Simulation study

A Monte Carlo simulation study was conducted to test the ability of the new method to recover correct partial HGTs. We examined how the proposed method performs depending on the number of observed species and number of generated partial transfers. First of all, we calculated the distribution of median gene sizes of prokaryotic genomes (Figure 2) considering 1494 complete microbial genomes available in the GenBank database in April 2011 (for more details, see: http://www.ncbi.nlm.nih.gov/genomes/lproks.cgi). The 'average median gene size' (i.e. the average here was taken over all calculated median gene sizes; Figure 2) of a microbial genome was 268 amino acids (the standard deviation was equal to 24 and the average size of a prokaryotic gene was 315 amino acids). The determined average median gene size of a prokaryotic genome was then used as a benchmark for MSA length in our simulations. Mention that the size of a transferred DNA fragment varies from organism to organism, and can be, in some situations, larger than a single gene [e.g. it is in the range 5–10 kb for some pathogenic bacteria; (41)]. Such longer

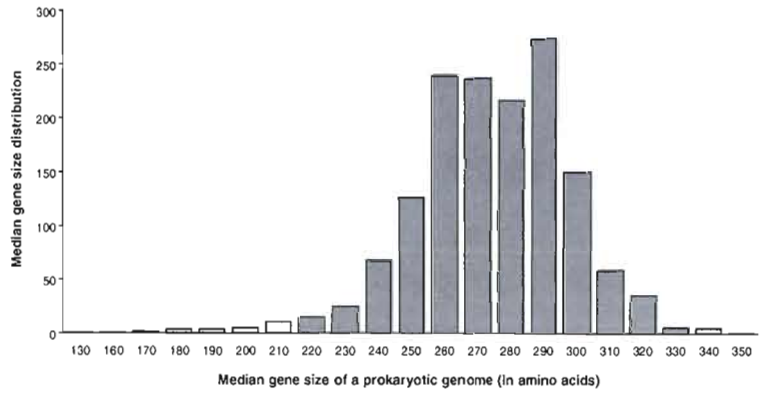


Figure 2. Distribution of median gene sizes of prokaryotic genomes computed on the basis of 1494 complete microbial genomes available in April 2011 in the GenBank database [(46), for more details, see: http://www.ncbi.nlm.nih.gov/genomes/lproks.cgi]. The height of each column corresponding to the graduation mark k of the abscissa axis represents the number of genomes whose median gene sizes are located in the interval $[k\text{-}5; k+4]$. For instance, the column corresponding to the mark 300 of the abscissa axis accounts for the genomes whose median gene sizes comprise between 295 and 304 amino acids.

transferred fragments can be detected using the new method by treating the complete genomes of involved organisms on the gene-by-gene basis. The most significant among the obtained partial transfers can then be merged to form longer sequence segments affected by HGT Furthermore, any existing method for the identification of complete HGTs (e.g. LatTrans, RIATA–HGT, HGT-Detection) can be used to confirm or discard complete HGTs detected by our method.

The simulation protocol included four main steps described below. First, random binary species trees with 8, 16, 32 and 64 leaves were generated using the procedure described by Kuhner and Felsenstein (42). The branch lengths of the species trees were generated using an exponential distribution. Following the approach of Guindon and Gascuel (43), we added some noise to the branches of the species phylogenies in order to provide a deviation from the molecular clock hypothesis. The trees yielded by this procedure had depth of $O[\log(n)]$, where n is the number of species (i.e. number of leaves in a binary phylogenetic tree).

Second, we carried out the SeqGen program (44) to generate random multiple sequence alignments of protein sequenees along the branches of the species trees constructed at the first step. The SeqGen program was used with the JTT model of proteins substitution (45). Protein sequences with 268 amino acids (i.e. average median gene size of a prokaryotic genome) were generated.

Third, having the sequences corresponding to the nodes of each species tree T, we, in turn, generated gene trees with the same number of leaves by performing random SPR (Subtree Prune and Regraft) moves of its subtrees. A model satisfying all plausible evolutionary constraints was implemented to generate random HGTs. For each species tree, 1–4 random SPR moves were performed and different gene trees T', encompassing 1–4 partial HGT events, were created. For each gene tree, the sequence fragments involved in the transfer(s) were identified and the corresponding sequence(s) in the subtree(s) affected by HGT were regenerated using SeqGen. Two different sizes of transferred fragments, 89 and 134 amino acids, corresponding respectively to one-third and one-half of the total gene length, were tested in our simulations. The tests conducted with two different transferred fragment sizes were carried out separately. When more than one HGT was generated, the sequence fragments affected by HGT could overlap. Thus, the obtained MSAs, each MSA included the sequences corresponding to the leaves of a gene tree, comprised blocks of amino acids affected by HGT.

Fourth, we carried out the new method for each generated species tree and the associated set of MSAs affected by partial HGT(s). The size of the sliding window was set to 100 sites; 100 replicates of each partial gene tree T' were generated to assess the bootstrap support of its branches, first, and the support of the obtained partial transfers, second. Partial gene trees whose average bootstrap support was <60% were ruled out from the analysis. Among the obtained HGTs only the transfers with bootstrap scores of 90% and higher were considered as significant and retained in the final solution.

Finally, we estimated the detection rate (i.e. true positives) and the false positive rate depending on the number of species and generated transfers. The obtained average performances of the new method are illustrated in Figures 3 and 4. For each set of parameters (tree size; number of generated HGTs), 100 replicated data sets were generated. On the other hand, Figure 5 highlights the difference between the average detection rate and average false positive rate with respect to the number of species. Figures 3 and 5a show that the best detection rates for the transferred fragments of size 89 amino acids were obtained for the 16-species trees. The results vary from 100% for one transfer to 69% for four transfers, giving a 79.9% partial HGT recovery on average. The best average false positive rate of 29.2% was obtained for the 32-species trees (Figure 5a). For the transferred fragments of size 134 amino acids, the best results were obtained for the 64-species trees (Figure 4). The average partial HGT detection rate for this size of trees was 81.1% and average false positive rate was 30.2% (Figure 5b). The average here was computed from the results obtained for 1–4 generated HGTs. Similar trends can be observed for the other tree sizes. According to our additional tests, these results can be improved by adjusting the simulation parameters with respect to the nature of the studied sequences.

Mention that high false positive rate obtained for the small trees (i.e. with 8 and 16 leaves) was mainly due to the fact that multiple minimum-cost HGT scenarios (i.e. solutions including the same minimum number of transfers) often exist in the case when small phylogenies are affected by several (e.g. 3 or 4) transfers (6, 20). For instance, Figure OA6 (e) in (6) shows that in case of complete gene transfers, we have only up to 40% of chances to obtain the same (correct) HGT scenario for 10-species trees and up to 47% for 20-species trees (the results in Figure OA6 are shown for the HGT-Detection and LatTrans algorithms). In order to lower the false positive rate that is higher for smaller trees (Figure 5), we conducted an additional simulation. Note that the results presented in Figures 3–5 correspond to the strategy in which any transfer with bootstrap scores of 90% and higher found for 'at least one fixed window position' was considered as significant. Such a strategy allows for a high hit detection rate but is also capable of generating some false positives transfers. We also considered algorithmic strategies where an HGT was recognized as significant if and only if it was found for 'at least 2, 3, 4 or 5 consecutive fixed window positions'. Such consecutive windows were overlapping each other because the window progress step of 10 sites was used in our simulations. Figure 6 illustrates the evolution of the average HGT detection rate (grey columns) and average false positive rate (white columns) depending on the number of consecutive windows for all of which the same transfer was detected. The averages here were taken over the results obtained for all considered trees sizes (8, 16, 32 and 64-species trees) and 1, 2, 3 and 4 generated HGTs; 100 trees were generated and tested for each combination of these parameters. The results presented in Figure 6 suggest that the strategy considering several consecutive windows can be effective for decreasing the false positive rate, especially in the case of longer transferred fragments (i.e.

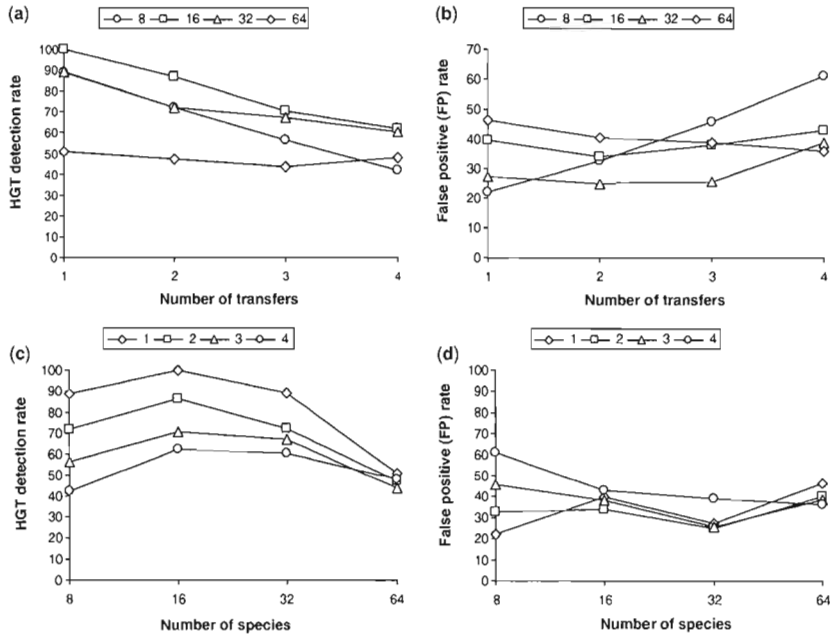


Figure 3. Average HGT detection rate depending on the number of transfers (a), and number of species (c). Average false positive (FP) rate depending on the number of transfers (b), and number of species (d). Each reported value represents the average result obtained for random trees with 8, 16, 32 and 64 leaves (cases a and b), and 1–4 HGTs (cases c and d); 100 replicates were generated for each parameter combination. The presented results were obtained with the sequences of length 268 amino acids (i.e. median prokaryotic genome gene size) and HGT fragments of 89 amino acids (i.e. one-third of the total gene length).

134 amino acids). Certainly, the improvement in the false positives rate was obtained at the expense of the detection rate. For the transferred fragments of 89 amino acids, the false positive rate decreased from 37.6% to 21.3%, while for the 134 amino acids fragments, it decreased from 39.4% to 19.1%, for one and five consecutive windows, respectively. The largest difference between the average false positive and false negative rates was obtained with one window, for the transferred fragments of 89 amino acids (31.3%), and with three consecutive windows for the fragments of 134 amino acids (44.0%). The lowest average false positive rate of 5.3% was obtained, while considering five consecutive windows, for 64-species trees and 134 amino acids fragments. This means that for longer transferred sequences and larger trees one should look for a result confirmation over a few consecutive window positions in order to validate the obtained transfers. The option allowing for validating the obtained HGTs for a series of consecutive window positions was included in our software available at: http://www.trex.uqam.ca.

The presented simulation results suggest that the new method can be useful for detecting partial transfers, and thus for identifying mosaic genes, especially when large trees and long sequence fragments affected by HGT are considered. With smaller transferred sequence fragments (i.e. one third of the total gene length), the best HGT detection rates were found for the trees with 16 and 32 leaves, whereas with larger transferred fragments (i.e. one-half of the total gene length), the best results were obtained for 64-leaf trees. While, on average, the HGT detection rates obtained for partial HGTs were slightly lower than those obtained by the LatTrans (20) and HGT-Detection algorithms for complete gene transfers [see Figure OA6 in Ref. (6)], we should notice that the problem of detecting partial HGTs is much more complex than the problem of inferring complete gene transfers. This complexity is due, first, to high similarity of short sequence fragments located in the sequence blocks affected by HGT and, second, to a possible overlap of the latter blocks which can disguise real gene transfers.

Detecting partial transfers of the gene rbcL

First, we applied the new method to analyze the Proteobacteria, Cyanobacteria and plastid data originally

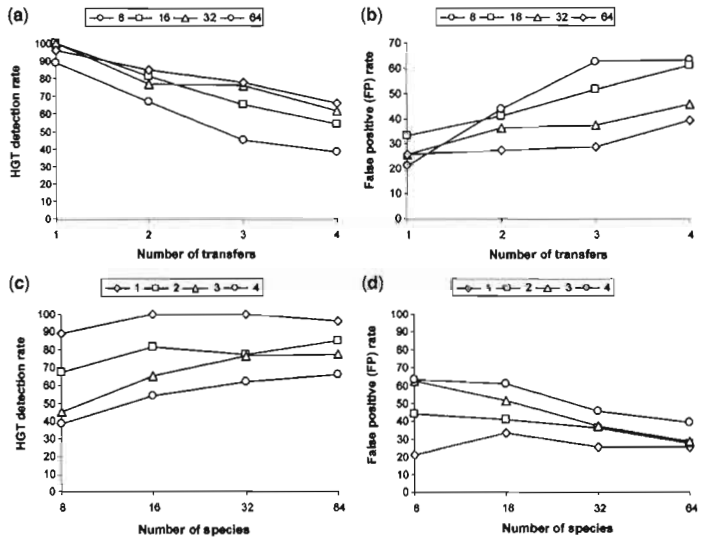


Figure 4. Average HGT detection rate depending on the number of transfers (**a**), and number of species (**c**). Average false positive (FP) rate depending on the number of transfers (**b**), and number of species (**d**). Each reported value represents the average result obtained for random trees with 8, 16, 32 and 64 leaves (cases a and b), and 1–4 HGTs (cases c and d); 100 replicates were generated for each parameter combination. The presented results were obtained with the sequences of length 268 amino acids (i.e. median prokaryotic genome gene size) and HGT fragments of 134 amino acids (i.e. one-half of the total gene length).

examined by Delwiche and Palmer (37). The latter authors discussed the hypothesis of HGT of the rubisco genes versus the hypothesis of ancient gene duplication followed by partial gene loss. Delwiche and Palmer (37) inferred a maximum parsimony phylogeny of the gene *rbcL* (large subunit of rubisco) for 48 organisms, including 42 taxa for Form I and 6 taxa for Form II of rubisco. They pointed out that the classification based on the gene *rbcL* contained numerous conflicts compared to the classification based on 16S ribosomal RNA and other evidence. The aligned *rbcL* amino acid sequences comprising 532 bp considered by Delwiche and Palmer and reanalyzed in this study can be found at: http://www.life.umd.edu/labs/delwiche.

To perform the analysis, we retained 42 of 48 organisms from the original study: all the taxa of Form I of *rbcL* were examined, whereas the 6 taxa of Form II, used by Delwiche and Palmer (37) to root the gene tree, were discarded. For the species *Chromatium* and *Hydrogenovibrio* two different copies of the rubisco gene, denoted, respectively, *Chromatium A* and *L*, and *Hydrogenovibrio L1* and *L2*, were considered in the original study. Thus, in this example, the gene phylogeny comprised 42 organisms, while the species phylogeny only 40. It is worth noting that the new method was adapted to the case when the species and gene trees have different number of leaves. The ML tree of the gene *rbcL* inferred using the PhyML method (39) is shown in Figure 7. This tree is very similar to the maximum parsimony gene tree obtained by Delwiche and Palmer (see Figure 2 in Ref. 37). The organisms *Pseudomonas* and endosymbiont of *Alviniconcha*, denoted as uncertain in Figure 2 of Delwiche and Palmer (37), were later classified as β-proteobacteria.

The corresponding species tree (Figure 8, undirected branches) was reconstructed and rooted using the appropriate information from the NCBI taxonomic browser (46). Since in this study we were mostly interested in identifying the transfers between different groups of organisms, we deliberately kept intact in the species tree, with respect to the topology of the gene tree, the positions of the organisms belonging to the same group. For instance, the topologies of the clades of Green plastids, Cyanobacteria, and Red and Brown plastids were identical in the gene and species phylogenies shown in Figures 7 and 8, respectively. A number of important topological conflicts between the species and gene trees can be observed. For example, there exists a large clade in the gene tree with bootstrap support of 98% (Figure 6), including one α-proteobacterium, three β-proteobacteria, six γ-proteobacteria and one cyanobacterium. Such topological conflicts can be explained either by frequent HGT events (partial or complete) or by ancient gene duplication

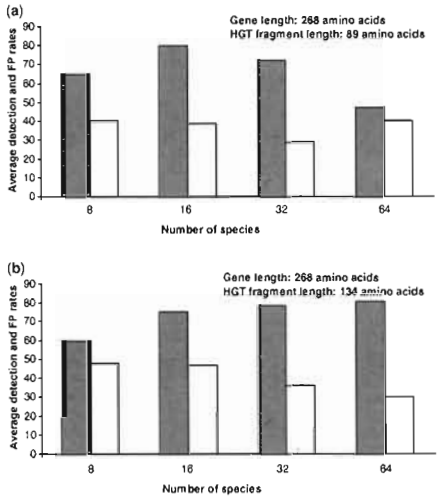


Figure 5. Average, over 1–4 generated transfers, HGT detection rate (grey columns) and false positive rate (white columns) depending on the number of species, obtained for the sequences of length 268 amino acids (i.e. median prokaryotic genome gene size) and HGT fragments of: (**a**) 89 amino acids (i.e. one-third of the total gene length) and (**b**) 134 amino acids (i.e. one-half of the total gene length).

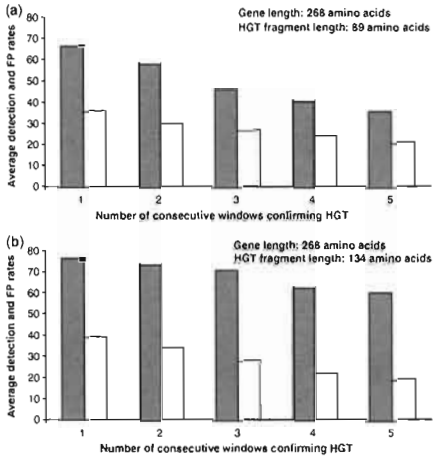


Figure 6. Average HGT detection rate (grey columns) and false positive rate (white columns), computed over 1–4 generated transfers and trees with 8, 16, 32 and 64 leaves, depending on the number of consecutive windows confirming the same HGT, obtained for the sequences of length 268 amino acids (i.e. median prokaryotic genome gene size) and HGT fragments of: (**a**) 89 amino acids (i.e one-third of the total gene length) and (**b**) 134 amino acids (i.e. one-half of the total gene length).

followed by gene losses [these two hypotheses are not mutually exclusive; see reference (37) for more details]. Below, we consider only the HGT hypothesis to explain topological incongruence between the species and gene phylogenies.

First, we carried out the HGT-Detection method for predicting complete HGTs (6); the bipartition dissimilarity criterion was used for optimization. The minimum-cost transfer scenario with nine HGTs necessary to reconcile the species and gene phylogenies is shown in Figure 8 (HGTs are depicted by numbered arrows). The optimality of this solution was confirmed by the LatTrans algorithm (20) based on the exhaustive search. The bootstrap support of the obtained complete HGTs was also computed.

Second, we carried out the new method for predicting partial HGTs. We used the sliding windows of the size 200, 300 and 400 sites with the progress step of 10 sites. Partial trees corresponding to the subsequences located within the sliding window were inferred using the PhyML method (39) with the JTT model of proteins substitution (45). For the windows smaller than 200 sites, the average bootstrap score of the branches of partial trees was often smaller than 50% because of the strong similarity between the examined amino acid sequences. The HGT-Detection method (6) with the bipartition dissimilarity option was then performed to infer partial HGTs for each position of the sliding window. As a final result, we retained 10 partial transfers illustrated in Figure 9 (all partial transfers with bootstrap scores lower than 60% were discarded). Some of these transfers were indeed complete transfers.

Thus, the proposed technique for inferring partial HGTs allowed us to refine the results of a method predicting complete transfers. Some of the detected complete HGTs were confirmed (i.e. HGTs 2, 6 and 9), some of them were discarded (i.e. HGTs 5 and 8 with low bootstrap support), and some of them were reclassified as partial transfers (i.e. HGTs 1, 3, 4 and 7). Moreover, the three new (partial) HGTs were found (i.e. HGTs 10, 11 and 12). For instance, the *rbcL* gene of *Chromatium L* is composed of the sequence polymorphisms stemming from *Hydrogenovibrio L1* (on the interval 130:230) and *L2* (on the interval 361:531) as well as from the original sequence (on the intervals 1:129 and 231:360). Obviously, the bootstrap scores of partial transfers, found for a part of the MSA, were higher than the corresponding bootstrap scores of complete transfers, found for the whole MSA.

The transfers shown in Figures 8 and 9 include one of the main HGTs predicted by Delwiche and Palmer [see Figure 4 in Ref. (37) and the following discussion]: Between α-proteobacteria and Red and Brown algae (complete HGT 2 with bootstrap support of 83.2%). The exact transfer between Cyanobacteria and the ancestor of β- and γ-proteobacteria (complete HGT 9

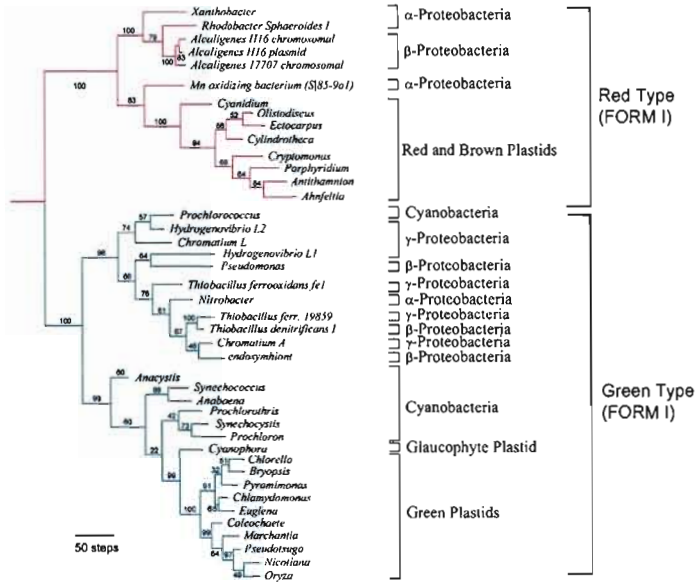


Figure 7. ML tree of the gene *rbcL* for 42 bacteria and plastids inferred from the rubisco amino acid sequences with 532 bases using the PhyML method (39). Taxa classification based on 16S rRNA and other evidence is indicated to the right. Numbers above the branches are their bootstrap scores calculated using 100 replicates.

with 87.1% bootstrap support) was not predicted by Delwiche and Palmer (37), but the latter study discussed the possibility of a close ancient transfer between Cyanobacteria and the ancestor of γ-proteobacteria. The obtained partial HGT scenario does not include, however, any HGT from γ-proteobacteria to α- and β-proteobacteria hypothesized by Delwiche and Palmer (37). To resolve multiple topological conflicts between the species and gene phylogenies, our scenario relies on HGTs from β-proteobacteria to α- and γ-proteobacteria, and from α- to β-proteobacteria.

Detecting partial transfers of the gene mutU

Second, we examined the evolution of the bacterial mismatch repair (MMR) gene *mutU* of *Escherichia coli* originally discussed by Denamur *et al.* (30). Denamur *et al.* explored the hypothesis that MMR deficiency emerging in nature has left some 'imprint' in the bacterial genomes and showed that individual functional MMR genes, when compared to housekeeping genes, exhibit high sequence mosaicism derived from different phylogenetic lineages. The *E. coli* MMR genes, *mutS*, *mutL*, *mutH* and *mutU* (*uvrD*), and two control genes (*mutT* and *recD*), were partially sequenced from 30 natural isolates in order to test the transfer hypothesis. Denamur *et al.* (30) compared the obtained gene phylogenies to the whole genome reference tree and found numerous topological conflicts that ranged from single (for *mutT*) to multiple (for *mutS*). To test whether these topological conflicts were due to HGT or tree reconstruction artefacts, the latter authors applied the incongruence length difference (ILD) method (47) and concluded that the MMR gene trees, when compared to the whole genome tree, exhibit significant incongruence due, most likely, to horizontal gene transfer. Supplementary Figure S1 reports the hypothetical partial horizontal transfers of the gene *mutU* within the *E. coli* evolutionary tree found in Ref. (30). Because of the highest level of mosaicism within MMR genes, the strain ECOR 37 does not have a clear phylogenetic position within the *E. coli* stain phylogeny (see Supplementary Figure S1, where this strain is not included in the set of tree leaves).

The new method was applied on the MSA of the gene *mutU* MMR, using three different window sizes: 100, 150 and 200 sites and the advancement step of 10 sites. The total length of the *mutU* MSA was 384 nt. The aligned sequences of the gene *mutU* that we examined can be found at: http://www.info2.uqam.ca/~makarenkov_v/mutU.txt. To build the *mutU* tree, we used the HKY85 (48) substitution model and the default settings of PhyML. Because of the strong similarity between the

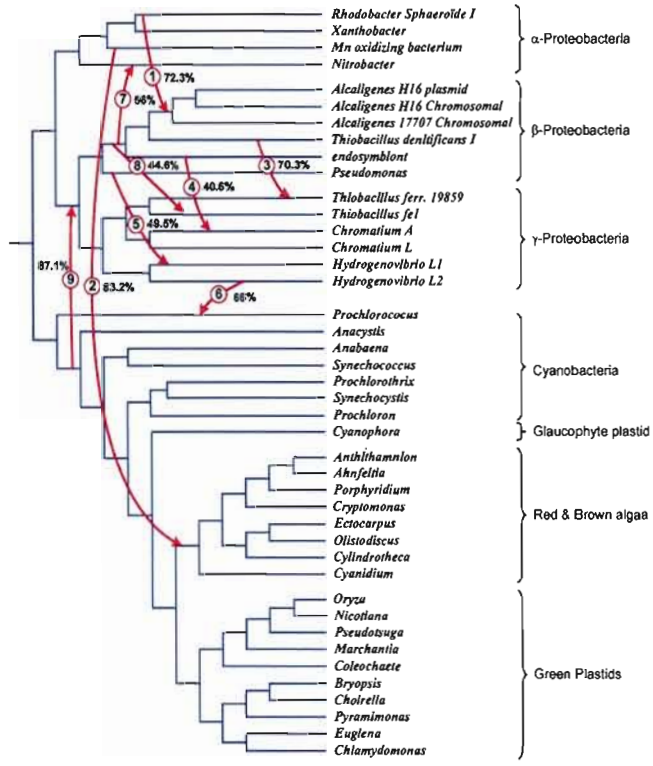


Figure 8. Species tree for the 42 bacteria and plastid organisms from Figure 7 with 9 HGT branches (denoted by arrows) representing complete horizontal transfers of the gene *rbcL*. This scenario was a unique shortest complete HGT scenario found for the given pair of species and gene trees. Bootstrap support of complete HGT events is indicated.

DNA sequences, multiple unresolved partial gene trees were found. All partial gene trees whose average bootstrap score was under 50% were ruled out from the analysis (i.e. not treated by the HGT detection method).

Figure 10 presents the eight most significant transfers inferred by the new method (the transfers whose bootstrap support was greater than 40% are represented). For each transfer, its direction, involved species, bootstrap support and associated interval of the original MSA are depicted.

For instance, HGTs 1, 3 and 4, with bootstrap support of 60%, 65% and 46%, respectively, correspond to three similar transfers found by Denamur *et al.* (30), Supplementary Figure S2; in the latter study, an exact analogue of HGT 4 was not determined, but a very close transfer was found. HGT 2 detected by the new method was also identified by Denamur *et al.* (30), but it goes in the opposite direction in that study. It is worth noting that all eight transfers found by Denamur *et al.* (30) were also predicted by the new method, but four of them are not represented in Figure 10 as a consequence of their low bootstrap support. We also found four new partial gene transfers (HGTs 5, 6, 7 and 8) with high bootstrap scores (63%, 94%, 75% and 70%, respectively). Mention that the solution found by the HGT-Detection method for inferring complete transfers (6) included only HGTs 2 and 3 from Figure 10. The other transfers found by HGT-Detection were different from those represented in Figure 10 and usually had a low bootstrap support.

DISCUSSION

We described a new method for predicting partial HGT events followed by intragenic recombination and thus for

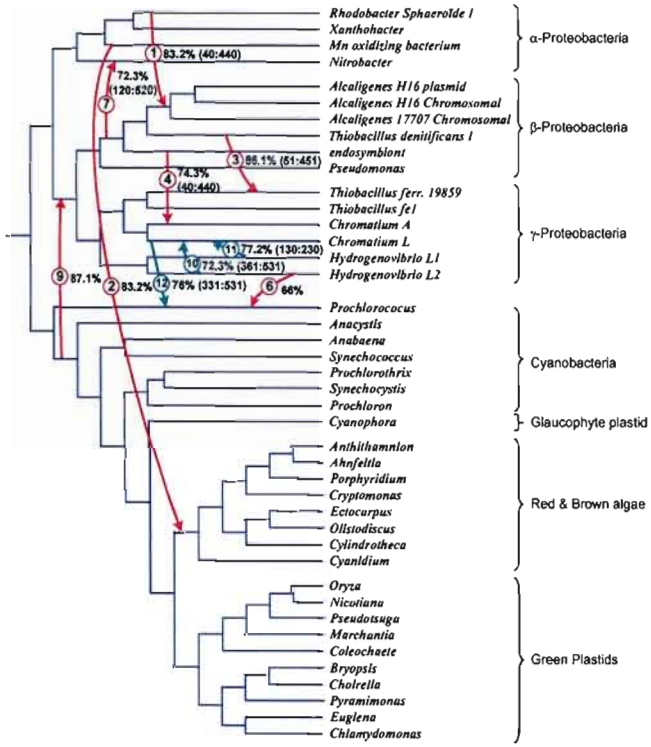


Figure 9. Species tree for the 42 bacteria and plastids from Figure 7 with 10 HGT branches (denoted by arrows) representing partial horizontal transfers of the gene *rbcL*. Partial HGTs having their analogs among complete HGTs received the same numbers as in Figure 8; HGTs absent in Figure 8 are numbered 10–12. Bootstrap support of partial HGT events and affected intervals of the original MSA are indicated. For complete HGTs 2, 6 and 9 (affecting the whole MSA) the interval is not indicated

identifying the origins of mosaic genes. To the best of our knowledge, this relevant problem has not been properly addressed in the literature [for instance, the two existing partial HGT detection methods, (30) and (31), do not include any validation of the obtained gene transfers or Monte Carlo simulations]. The proposed method is based on a sliding window procedure that progressively analyzes the fragments of the given sequence alignment. The size of the sliding window should be adjusted with respect to the existing information about the genes and species under study. The use of smaller sizes of the sliding window allows one to detect smaller partial transfers with a better accuracy (i.e. HGTs affecting shorter intervals of the given multiple sequence alignment), but this also increases the running time of the method. For each fixed window position, a corresponding partial tree is inferred and a scenario of partial HGT events is determined by reconciling the obtained partial gene tree and the given species tree. A bootstrap procedure, allowing one to assess the bootstrap support of partial HGTs by taking into account the uncertainty of partial gene trees, was also developed. Another advantage of the presented method over the existing sliding window techniques used to detect recombination (33–37) is that it also allows for detecting the source (i.e. from which donor species the transferred fragments arrived) of the transferred sequences. The described method was included in the T-REX package (49) available at: http://www.trex.uqam.ca.

Both examples considered in the 'Results' section suggest that the new method can be also useful for confirming or discarding complete HGTs inferred by any existing HGT detection method. Our study of the evolution of the gene *rbcL* for 40 species of Proteobacteria, Cyanobacteria and plastids (37) and that of the mismatch repair (MMR) gene *mutU* for 30 *E. coli* strains (30) showed that most of the predicted gene

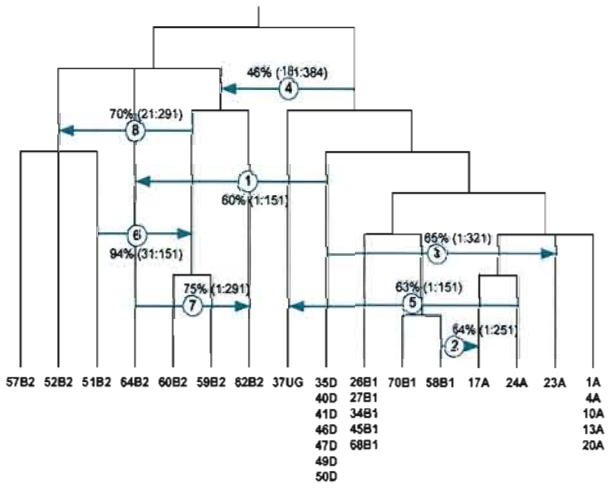


Figure 10. Hypothetical partial transfers of the gene *mutU* predicted by the new method. Partial HGTs are denoted by arrows. Bootstrap support of partial HGT events and affected intervals of the original MSA are indicated.

transfers (i.e. six out of eight for each data set) may have been, in fact, partial HGTs. The conducted simulations showed that with smaller transferred sequence fragments, the best HGT detection rates were found for the trees having 16 and 32 leaves, whereas with larger transferred fragments the best results were obtained for 64-leaf trees. The following general trend can be formulated when analyzing the results presented in Figures 3–5: longer transferred sequence fragments and larger trees provide a much better HGT recovery and a smaller number of false positives. The problem occurring when considering short sequence fragments is that partial phylogenies inferred from them usually have low bootstrap support, and consequently provide a low confidence level of detected HGTs. The simulation results also suggest that in case of longer transferred sequences and larger trees one should look for a result confirmation over a few consecutive window positions in order to validate the obtained transfers.

The results of crosses with either the same donor or the same recipient show that recombination frequency decreases exponentially with increasing sequence divergence (50). Thus, the recombination success is strongly dependent on percent of nucleotide identity, which implies that recombination breakpoints occur only in the most conserved parts of a gene. This feature can be integrated into the described method by considering a more comprehensive statistical model taking into account the sequence divergence parameter. On the other hand, information about the obtained partial HGTs and their bootstrap scores can be incorporated in an extended evolutionary model that takes into account horizontal gene transfer, ancient gene duplication and gene loss (e.g. topological incongruence giving rise to predicted partial and/or complete transfers with low bootstrap support may be due to ancient gene duplication followed by partial gene loss). The determined bounds of transferred fragments can be examined in more details by comparing the corresponding 3D conformations. The discussed method can be also applied on a full-genome scale to estimate the proportion of mosaic genes in each studied genome as well as the rates of partial and complete HGTs between involved species. Several relevant statistics regarding the position and functionality of genetic fragments affected by horizontal gene transfer along with the rates of intraspecies (i.e. HGT between strains of the same species) and interspecies (i.e. HGT between distinct species) transfers can be estimated using the discussed technique. An alternative approach that can be also envisaged would be based on a Hidden Markov Model applied along the given MSA with the hidden state representing the HGT history of each considered sequence fragment. As any method of phylogenetic analysis, the presented algorithm for detecting partial gene transfers is subject to some artifacts. The main of them are long-branch attraction, unequal evolutionary rates and situations when possible HGT events almost coincide with speciation events. In the future, it will be important to investigate the impact of these artifacts on the identification of mosaic genes.

SUPPLEMENTARY DATA

Supplementary Data are available at NAR Online.

ACKNOWLEDGEMENTS

The authors thank Dr Hervé Philippe and two anonymous reviewers for their helpful comments.

FUNDING

Funding for open access charge: Natural Sciences and Engineering Research Council of Canada (NSERC); Nature and Technologies Research Funds of Quebec (FQRNT).

Conflict of interest statement. None declared.

REFERENCES

1. Doolittle,W.F. (1999) Phylogenetic classification and the universal tree. *Science*, **284**, 2124–2129.
2. Koonin,E.V. (2003) Horizontal gene transfer: the path to maturity. *Mol. Microbiol.*, **50**, 725–727.
3. Doolittle,W.F., Boucher,Y., Nesbø,C.L., Douady,C.J., Anderson,J.O. and Roger,A.J. (2003) How big is the iceberg of which organellar genes in nuclear genomes are but the tip? *Philos. Trans. R. Soc. Lond. B. Biol Sci.*, **358**, 39 57.
4. Nakhleh,L., Ruths,D. and Wang,L.S. (2005) RIATA-HGT: a fast and accurate heuristic for reconstructing horizontal gene transfer. In Wang,L. (ed.), *Lecture Notes in Computer Science*. Springer, Kunming, China, pp. 84–93.
5. Makarenkov,V., Kevorkov,D and Legendre,P. (2006) Phylogenetic network reconstruction approaches. *Bioinformatics*, **6**, 61–97.
6. Boc,A., Philippe,H. and Makarenkov,V. (2010) Inferring and validating horizontal gene transfer events using bipartition dissimilarity. *Syst. Biol.*, **59**, 195–211.
7. Hollingshead,S.K., Becker,R. and Briles,D.E. (2000) Diversity of PspA: mosaic genes and evidence for past recombination in *Streptococcus pneumoniae*. *Infect. Immun.*, **68**, 5889 5900.
8. Zhaxybayeva,O., Lapierre,P. and Gogarten,J.P. (2004) Genome mosaicism and organismal lineages. *Trends Genet.*, **20**, 254–260.
9. Gogarten,J.P., Doolittle,W.F. and Lawrence,J.G. (2002) Prokaryotic evolution in light of gene transfer. *Mol. Biol. Evol.*, **19**, 2226–2238.
10. Lawrence,J.G. and Ochman,H. (1997) Amelioration of bacterial genomes: rates of change and exchange. *J. Mol. Evol.*, **44**, 383–397.
11. Maiden,M. (1998) Horizontal genetic exchange, evolution, and spread of antibiotic resistance in bacteria. *Clin. Infect. Dis.*, **27**, 12–20.
12. Zheng,Y., Roberts,R.J. and Kasif,S. (2004) Segmentally variable genes: a new perspective on adaptation. *PLoS Biol.*, **2**, 452–464.
13. Claverys,J.P., Prudhomme,M., Mortier-Barrière,I. and Martin,B. (2000) Adaptation to the environment: *Streptococcus pneumoniae*, a paradigm for recombination-mediated genetic plasticity? *Mol. Microbiol.*, **35**, 251 259.
14. Hein,J. (1993) A heuristic method to reconstructing the evolution of sequences subject to recombination using parsimony. *J. Mol. Evol.*, **36**, 396–405.
15. von Haeseler,A. and Churchill,G.A. (1993) Network models for sequence evolution. *J. Mol. Evol.*, **37**, 77–85.
16. Page,R.D.M. (1994) Maps between trees and cladistic analysis of historical associations among genes, organism and areas *Syst. Biol.*, **43**, 58–77.
17. Mirkin,B.G., Muchnik,I. and Smith,T.F. (1995) A biologically consistent model for comparing molecular phylogenies. *J. Comput. Biol.*, **2**, 493–507.
18. Maddison,W.P. (1997) Gene trees in species trees. *Syst. Biol.*, **46**, 523–536.
19. Charleston,M.A. (1998) Jungle: a new solution to the host/ parasite phylogeny reconciliation problem. *Math. Biosci.*, **149**, 191–223.
20. Hallett,M. and Lagergren,J. (2001) Efficient algorithms for lateral gene transfer problems. In El-Mabrouk,N., Lengauer,T. and Sankoff,D. (eds). *Proceedings of the Fifth Annual International Conference on Research in Computational Biology*. ACM Press, New York, pp. 149–156.
21. Boc,A. and Makarenkov,V. (2003) New efficient algorithm for detection of horizontal gene transfer events. In Benson,G. and Page,R. (eds), *Algorithms in Bioinformatics*. Springer, Budapest, Hungary, pp. 190–201.
22. MacLeod,D., Charlebois,R.L., Doolittle,F. and Bapteste,E. (2005) Deduction of probable events of lateral gene transfer through comparison of phylogenetic trees by recursive consolidation and rearrangement. *BMC Evol. Biol.*, **5**, 27
23. Tsirigos,A. and Rigoutsos,I. (2005) A new computational method for the detection of horizontal gene transfer events. *Nucleic Acids Res.*, 33, 922 933.
24. Tsirigos,A. and Rigoutsos,I. (2005) A sensitive, support-vector-machine method for the detection of horizontal gene transfers in viral, archaeal and bacterial genomes. *Nucleic Acids Res.*, **33**, 3699–3707.
25. Beiko,R.G. and Hamilton,N. (2006) Phylogenetic identification of lateral genetic transfer events *BMC Evol. Biol.*, **6**, 15.
26. Jin,G., Nakhleh,L., Snir,S. and Tuller,T. (2006) Efficient parsimony-based methods for phylogenetic network reconstruction. *Bioinformatics*, **23**, 123–128.
27. Jin,G., Nakhleh,L., Snir,S. and Tuller,T. (2007) Inferring phylogenetic networks by the maximum parsimony criterion. *Mol Biol. Evol.*, **241**, 324–337.
28. Linz,S., Radtke,A. and von Haeseler,A. (2007) A maximum likelihood framework to measure horizontal gene transfer. *Mol. Biol. Evol.*, **24**, 1312 1319.
29 Than,C. and Nakhleh,L. (2008) SPR-based tree reconciliation: non-binary trees and multiple solutions. In *Proceedings of the 6th Asia Pacific Bioinformatics Conference*. Kyoto, Japan, pp. 251–260
30. Denamur,E., Lecointre,G., Darlu,P , Tenaillon,O., Acquaviva,C., Sayada,C., Sunjevaric,I., Rothstein,R., Elion,J., Taddei,F. *et al.* (2000) Evolutionary implications of the frequent horizontal transfer of mismatch repair genes *Cell.*, **103**, 711–721.
31. Makarenkov,V., Boc,A., Delwiche,C.F., Diallo,A.B. and Philippe,H. (2006) New efficient algorithm for modeling partial and complete gene transfer scenarios. In Batagelj,V , Bock,H H., Ferligoj,A and Ziberna,A. (eds), *Data Science and Classification*. Springer, pp. 341–349.
32. Ray,S.C (1998) SimPlot for Windows (version 1.6). Berlin, Germany, Baltimore, Md.
33. Archibald,J.M. and Roger,A.J. (2002) Gene duplication and gene conversion shape the evolution of archaeal chaperonins. *J. Mol. Biol.*, **316**, 1041–1050.
34. Paraskevis,D., Deforche,K., Lemey,K., Magiorkinis., Hatzakis,A. and Vandamme,A.M. (2005) SlidingBayes: exploring recombination using a sliding window approach based on Bayesian phylogenetic inference. *Bioinformatics*, **21**, 1274–1275.
35. Lee,W.H. and Sung,W.K. (2008) RB-finder: an improved distance-based sliding window method to detect recombination breakpoints. *J. Comput. Biol.*, **15**, 881–898.
36. Martin,D.P., Lemey,P., Lott,M., Moulton,V., Posada,D. and Lefeuvre,P. (2010) RDP3: a flexible and fast computer program for analyzing recombination. *Bioinformatics*, **26**, 2462 2463
37. Delwiche,C.F. and Palmer,J.D. (1996) Rampant horizontal transfer and duplication of Rubisco genes in Eubacteria and Plastids. *Mol. Biol. Evol.*, **13**, 873–882.
38. Acinas,S.G., Marcelino,L.A., Klepac-Ceraj,V and Polz,M.F. (2004) Divergence and redundancy of 16S rRNA sequences in genomes with multiple rrn operons. *J. Bacteriol.*, **38**, 2629–2635.
39. Guindon,S. and Gascuel,O. (2003) A simple, fast and accurate algorithm to estimate large phylogenies by maximum likelihood. *Syst. Biol.*, **52**, 696 704.
40. Robinson,D.R. and Foulds,L.R. (1981) Comparison of phylogenetic trees. *Math. Biosciences*, **53**, 131–147.
41. Smith,J.M., Feil,E.J. and Smith,N.H. (2000) Population structure and evolutionary dynamics of pathogenic bacteria. *Bioessays*, **22**, 1115 1122.

42. Kuhner,M. and Felsenstein,J. (1994) A simulation comparison of phylogeny algorithms under equal and unequal evolutionary rates. *Mol. Biol. Evol.*, **11**, 459–468.
43. Guindon,S. and Gascuel,O. (2002) Efficient biased estimation of evolutionary distances when substitution rates vary across sites. *Mol. Biol. Evol.*, **19**, 534–543.
44. Rambaut,A. and Grassly,N.C. (1997) Seq-Gen: an application for the Monte Carlo simulation of DNA sequence evolution along phylogenetic trees. *Comput. Appl. Biosci.*, **13**, 235–238.
45. Jones,D.T., Taylor,W.R and Thornton,J.M. (1992) The rapid generation of mutation data matrices from protein sequences. *Comput. Appl. Biosci.*, **8**, 275–282.
46. Benson,D.A., Karsch-Mizrachi,I., Lipman,D.J., Ostell,J. and Sayers,E.W. (2009) GenBank. *Nucleic Acids Res.*, **37**, D26–D31.
47. Farris,J.S., Kallersjo,M., Kluge,A.G. and Bult,C. (1994) Testing significance of incongruence. *Cladistics*, **10**, 3, 315–319.
48. Hasegawa,M., Hirohisa,K. and Taka-aki,Y. (1985) Dating the human-ape splitting by a molecular clock of mitochondrial DNA. *J. Mol. Evol.*, **22**, 160–174.
49. Makarenkov,V. (2001) T-Rex: reconstructing and visualizing phylogenetic trees and reticulation networks. *Bioinformatics*, **17**, 664–668.
50. Vulic,M., Dionisio,F., Taddei,F. and Radman,M. (1997) Molecular keys to speciation: DNA polymorphism and the control of genetic exchange in enterobacteria. *Proc. Natl Acad. Sci. USA*, **94**, 9763–9767.

GLOSSAIRE

ADN (acide désoxyribonucléique) : macromolécule constituée de deux chaînes enroulées en double hélice. Ses deux brins sont assemblés à partir de nucléotides. Chaque nucléotide comprend un sucre, le désoxyribose, un phosphate et une des quatre bases azotées (adénine, guanine, cytosine et thymine). L'ADN est le support de l'information génétique des organismes vivants.

Alignement : opération qui consiste à disposer les unes en dessous des autres des portions de séquences similaires en minimisant leurs différences (on peut aligner entre eux des gènes d'une même famille multigénique ou des gènes d'espèces différentes). Si ces gènes sont homologues, les différences d'aminoacides ou d'acides nucléiques entre les séquences actuelles sont le témoignage de mutations qui ont eu lieu dans le passé.

Aminoacide (acide aminé) : unité constitutive des protéines. Il existe 20 acides aminés communs : alanine, arginine, asparagine, aspartate, cystéine, glutamine, glycine, histidine, isoleucine, leucine, lysine, méthionine, phénylalanine, proline, glutamate, sérine, thréonine, tryptophane, tyrosine et valine.

Archaea : les Archées ou *Archaea* (anciennement appelés archéobactéries, du grec *archaios*, « ancien » et *backterion*, « bâton ») sont un groupe majeur de microorganismes. Elles constituent un taxon du vivant caractérisé par des cellules sans noyau et se distinguant des Eubactéries (vraies bactéries) par certains caractères biochimiques, comme la constitution de la membrane cellulaire ou le mécanisme de réplication de l'ADN.

ARN (acide ribonucléique) : polymère linéaire dont la sous-unité de base, un ribonucléotide, contient le sucre ribose.

Bacteria : les bactéries appartiennent au vaste ensemble des microbes qui comprennent également les virus, les champignons et les parasites. Microorganismes invisibles à l'œil nu, les bactéries sont constituées d'une seule cellule dépourvue d'un vrai noyau. Elles contiennent un seul chromosome formé d'un long filament d'ADN.

Clade : vient du grec clados qui signifie arête. Taxon strictement monophylétique, c'est-à-dire contenant un ancêtre et tous ses descendants.

Cognat : Les **cognats**, ou **mots apparentés**, sont des mots qui ont une origine commune. Le terme peut désigner des mots d'une même langue, ou bien (le plus couramment) des mots dans des langues différentes. Par exemple, les mots *nuit* (en français), *night* (en anglais), et *nacht* (en allemand) sont apparentés, car ils sont issus d'une même racine indo-européenne. De même, les mots *père* et *paternel* sont apparentés, car tous deux issus du latin *pater*.

Eucarya : les Eucaryotes (du grec *eu*, vrai et *karuon*, noyau) comprennent 4 grands règnes du monde vivant : les animaux, les champignons, les plantes et les protistes. Ils constituent donc un très large groupe d'organismes, unis et pluricellulaires, définis par leur structure cellulaire (noyau, ADN, cytosquelette, etc).

Extragroupe (outgroup) : on dit aussi groupe extérieur ou encore "outgroup" tiré de l'anglais. Groupe que l'on sait *a priori* placé en dehors d'un ensemble de taxons dont on cherche les relations de parenté.

Horloge moléculaire (hypothèse) : l'hypothèse selon laquelle les molécules d'une même classe fonctionnelle évoluent régulièrement dans le temps à un rythme égal dans différentes lignées. Ainsi la quantité des différences moléculaires constatées de nos jours dans des séquences homologues d'espèces distinctes peut être utilisée pour estimer le temps écoulé depuis le dernier ancêtre commun à ces espèces (ou le temps de divergence).

Racine : le segment de arête en amont du nœud du rang le plus important, définissant le groupe extérieur (voir Extragroupe). En d'autres termes, c'est la position dans l'arbre du groupe extérieur. La racine peut être considérée comme un point de référence pour l'interprétation des caractères : les états de caractères de l'extragroupe (*outgroup*) sont des

états plésiomorphes, les états qui en diffèrent sont apomorphes. Remarque : pour pouvoir comparer aisément deux arbres, il faut les enraciner chacun avec la même espèce ou avec le même taxon.

Taxon : ensemble des organismes reconnus et définis dans chacune des catégories de la classification biologique hiérarchisée. En d'autres termes : contenu concret d'une catégorie. Exemple : *Canis lupus*, le loup, est un taxon de rang spécifique (catégorie : espèce) ; les canidés (Chien, Loup, Renard) constituent un taxon de rang familial (catégorie : famille).

BIBLIOGRAPHIE

Acinas, S.G., L.A. Marcelino, V. Klepac-Ceraj et M.F. Polz. (2004). Divergence and redundancy of 16S rRNA sequences in genomes with multiple rrn operons. *J. Bacteriol.*, volume 38, pages 2629-2635.

Adams K. L. et Palmer J. D. (2003). Evolution of mitochondrial gene content: gene loss and transfer to the nucleus. *Mol. Phylogenet. Evol.*, volume 29, pages 380–39

Addario-Berry, L., M. Hallett et J. Lagergren. (2003). Towards identifying lateral gene transfer events. *PSB*, volume 8, pages 279-290.

Allen, B.L. et M. Steel. (2001). Subtree transfer operations and their induced metrics on evolutionary trees. *Ann. Combin.*, volume 5, pages 1-15.

Andersson J.O. (2005). Lateral gene transfer in eukaryotes. *Cell Mol Life Sci.*, volume 62(11), pages 1182-97.

Atkinson, Q.D. et R.D. Gray. (2005). Curious Parallels and Curious Connections - Phylogenetic Thinking in Biology and Historical Linguistics. *Syst. Biol.*, volume 54, no. 4, pages 513-526.

Bandelt, H-J et A.W.M. Dress. (1989). Weak hierarchies associated with similarity measures – an additive clustering technique. *Bull. Math. Biol.*, volume 51, no. 1, pages 133-166.

Bandelt, H-J et A.W.M. Dress. (1992a). Split decomposition: a new and useful approach to phylogenetic analysis of distance data. *Mol. Phylogenet. Evol.*, volume 1, pages 242-252.

Bandelt, H-J et A.W.M. Dress. (1992b). A canonical decomposition theory for metrics on a finite set, *Adv in Math*, volume 92, pages 47-65.

Bandelt, H-J, P. Forster P, B.C. Sykes et M.B. Richards. (1995). Mitochondrial portraits of human populations using median networks. *Genetics*, volume 141, pages 743-753.

Bandelt, H-J, P. Forster et A Rohl. (1999). Median-joining networks for inferring intraspecific phylogenies. *Mol. Biol. Evol.*, volume 16, pages 37-48.

Bandelt, H-J, V. Macaulay et M. Richards. (2000). Median networks:speedy construction and greedy reduction, one simulation, and two case studies from human mtDNA. *Mol. Phylogenet. Evol.*, volume 16, pages 8-28.

Barthélemy, J.-P. et A.Guénoche. (1988). *Les arbres et les représentations des proximités*. Paris, Masson.

Barthelemy J.-P. et A. Guenoche. (1991). *Trees and proximity representations*. New York, Wiley.

Beiko, R. G., et N. Hamilton. (2006). Phylogenetic identification of lateral genetic transfer events. *BMC Evol. Biol.*, volume 6:15.

Boc, A. et V. Makarenkov. (2003). New Efficient Algorithm for Detection of Horizontal Gene Transfer Events, *Algorithms in Bioinformatics*, G. Benson et R. Page (Eds.), 3rd Annual WABI'03, Springer-Verlag, pages 190-201.

Boc, A., V. Makarenkov et A.B. Diallo. (2004). Une nouvelle méthode pour la détection de transferts horizontaux de gène : la réconciliation topologique d'arbres de gène et d'espèces, *JOBIM*, Montréal, Canada.

Boc, A., H. Philippe et V. Makarenkov. (2010a). Inferring and validating horizontal gene transfer events using bipartition dissimilarity, *Syst. Biol.*, volume 59, pages 195-211.

Boc, A., A-M. Di Sciullo et V. Makarenkov. (2010b). Classification of the Indo-European languages using a phylogenetic network approach. In *Classification as a Tool for Research*, H. Locarek-Junge et C. Weihs (Eds) proceedings of IFCS 2009. Studies in Classification, Data Analysis, and Knowledge Organization, Springer Berlin-Heidelberg-New York, pages 647-655.

Boc, A. et V. Makarenkov. (2011) Towards an accurate identification of partial horizontal gene transfers and intragénic recombination. *Nucl. Acids Res.*. Soumis.

Bordewich, M. et C. Semple. (2004). On the computational complexity of the rooted subtree prune and regraft distance. *Ann. Combin.*, volume 8, pages 409-423.

Bordewich, M., O. Gascuel, K. T. Huber et V. Moulton. (2009). Consistency of topological moves based on the balanced minimum evolution principle of phylogenetic inference. *IEEE/ACM Trans Comput Biol Bioinform*, volume 6, pages 110-117.

Bryant D., F. Filimon et R.D. Gray. (2005). Untangling our past: Languages, trees, splits and networks. In *The evolution of cultural diversity: Phylogenetic approaches*, R. Mace, C. Holden et S. Shennan (Eds.), London, UCL Press, pages 69-85.

Cannon, G. (1999). Problems in studying loans. *Proc. An. Meet. Berk. Ling. Soc.*, volume 25, pages 326-336.

Castresana, J. (2000). Selection of conserved blocks from multiple alignments for their use in phylogenetic analysis. *Mol. Biol. Evol.*, volume 17, pages 540-552.

Charleston, M.A. (1998). Jungle: a new solution to the host/parasite phylogeny reconciliation problem. *Math Biosc*, volume 149, pages 191-223.

Chernih, P. (2001). *Etymological dictionary of the modern Russian language*. Russki Jasik. 4th edition.

Comrie, B. (1981) *The Languages of the Soviet Union*. The Press Syndicate of the University of Cambridge.

Csürös, M., et I. Miklós. (2006). A probabilistic model for gene content evolution with duplication, loss, and horizontal transfer. In *Research in Computational Molecular Biology*, Springer Verlag, pages 206-220.

Bopp, F. (1867). *Comparative grammar of the Sanskrit, Zend, Greek, Latin, Lithuanian, Gothic, German, and Slavonic languages*. Translated principally by Lieutenant Eastwick, London, Madden & Malcolm, pages 1845-1856.

Bryant, D et V. Moulton. (2004). Neighbor-Net: an agglomerative method for the construction of phylogenetic networks. *Mol. Biol. Evol.*, volume 21, pages 255-265.

Croft, W. (2000) *Explaining language change: An evolutionary approach*. Harlow, Pearson Education.

Darwin, C. (1871). *The descent of man*. London, Murray.

Davison, J. (1999). Genetic exchange between bacteria in the environment. *Plasmid*, volume 42, pages 73-91.

Diamond, J. et P. Bellwood. (2003). Farmers and their languages: the first expansions. *Science*, volume 300, pages 597-603.

Diday, E. et P. Bertrand. (1984). An extension of hierarchical clustering: the pyramidal representation. In *Pattern Recognition in Practice*, E.S. Gelsema et L.N. Kanal (Eds.), Amsterdam, North-Holland, pages 411-424.

Delwiche, C.F. et J.D. Palmer. (1996). Rampant Horizontal Transfer and Duplication of Rubisco Genes in Eubacteria and Plastids. *Mol. Biol. Evol.*, volume 13, pages 873-882.

Denamur, E., G. Lecointre, P. Darlu, O. Tenaillon, C. Acquaviva, C. Sayada, I. Sunjevaric, R. Rothstein, J. Elion, F. Taddei, M. Radman et I. Matic (2000). Evolutionary implications of the frequent horizontal transfer of mismatch repair genes. *Cell*, volume 103, pages 711-721.

Doolittle W. F. (1998). You are what you eat: a gene transfer ratchet could account for bacterial genes in eukaryotic nuclear genomes. *Trends Genet.*, volume 14, pages 307–311.

Doolittle, W.F. (1999). Phylogenetic classification and the universal tree. *Science*, volume 284, pages 2124-2129.

Doolittle, W.F., Y. Boucher, C.L. Nesbo, C.J. Douady, J.O. Andersson et A.J. Roger. (2003). How big is the iceberg of which organellar genes in nuclear genomes are but the tip? *Phil. Trans. R. Soc. London B. Biol. Sci.*, pages 358:39-57.

Dyen, I., J.B. Kruskal, et P. Black. (1997). Fichier sur les langues Indo-Européennes (IE-DATA1) disponible à : http://www.ntu.edu.au/education/langs/ielex/IE-DATA1l.

Excoffier, L. et P.E. Smouse. (1994). Using allele frequencies and geographic subdivision to reconstruct gene trees within a species:molecular variance parsimony. *Genetics*, volume 136, pages 343-359.

Fasmer, M. *Russisches etymologisches*. Worterbuch, Heidelberg, pages 1950–1958.

Felsenstein, J. (1981). Evolutionary trees from DNA sequences: a maximum likelihood approach. *J. Mol. Evol.*, volume 17, pages 368-376.

Felsenstein, J. (1985). Confidence limits on phylogenies: An approach using the bootstrap. *Evolution*, volume 39, pages 738-791.

Felsenstein, J. (1989). PHYLIP - Phylogeny Inference Package (Version 3.2). *Cladistics*, volume 5, pages 164-166.

Fitch, W.M. (1997). Networks and viral evolution. *J Mol Evol*, volume 44, pages S65–S75.

Foulds, L.R., M.D. Hendy et D. Penny. (1979). A graph theoretic approach to the development of minimal phylogenetic trees. *J Mol Evol*, volume 13, pages 127-149.

Gauss, C.F. (1811). Disquisitio de Elementis Ellipticis Palladis. English translation of extract in pp. 148-155 of Trotter, H . F. (1957). Gauss's Work (1803-26) on the Theory of Least Squares, Technical Report 5, Statistical Techniques Research Group, Princeton University. A translation of Méthodes des Moindres Carrés, the authorised French translation of Gauss's writings on least squares by J. Bertrand (1855), Paris: Mallet-Bachelier.

Gogarten, J.P., W.F. Doolittle, et J.G. Lawrence. (2002). Prokaryotic Evolution in Light of Gene Transfer. *Mol. Biol. Evol.*, volume 19, pages 2226-2238.

Gogarten J. P. (2003). Gene transfer: gene swapping craze reaches eukaryotes. *Curr. Biol.*, volume 13, R53-R5.

Guindon, S. et O. Gascuel. (2002). Efficient biased estimation of evolutionary distances when substitution rates vary across sites. *Mol. Biol. Evol.*, volume 19, pages 534-543.

Guindon, S. et O. Gascuel. (2003). A simple, fast and accurate algorithm to estimate large phylogenies by maximum likelihood. *Syst. Biol.*, volume 52, pages 696-704.

Gray, R.D. et Q.D. Atkinson. (2003). Language-tree divergence times support the Anatolian theory of Indo-European origin. *Nature*, volume 426, pages 435-439.

Greek etymological dictionary. Disponible sur : <http://sdict.com >.

Hallett, M., et J. Lagergren. (2001). Efficient algorithms for lateral gene transfer problems.In *Proceedings of the fifth annual international conference on research in computational biology*, N. El-Mabrouk, T. Lengauer et D. Sankoff (Eds.), ACM Press, New-York, pages 149-156.

Hallett, M., J. Lagergren, et A. Tofigh. (2004). Simultaneous identification of duplications and lateral transfers. In *Proceedings of the eighth annual international conference on research in computational biology*, P.E. Bourne et D. Gusfield (Eds.), ACM, San Diego, pages 347-356.

Hamming R. (1950). E*rror-detecting and error-correcting codes. Bell System Technical Journal*, volume 29(2), pages 147-160.

Harper D. (accédé en 2010) *Online Etymology Dictionary* : http://www.etymonline.com.

Hasegawa, M., H. Kishino, et T. Yano. (1985). Dating of the human-ape splitting by a molecular clock of mitochondrial DNA. *J. Mol. Evol.*, volume 22(2), pages 160-174.

Haugen, E. (1950). The analysic of linguistic borrowing. *Language*, volume 26, pages 210-231.

Hein, J. (1990). A heuristic method to reconstructing the evolution of sequences subject to recombination using parsimony. *Math Biosci*, volume 98, pages 185-200.

Hein, J. (1993). A heuristic method to reconstruct the history of sequences subject to recombination, *J. Mol. Evol.*, volume 36, pages 369-405.

Hein, J., T. Jiang, L. Wang, et K. Zhang. (1996). On the complexity of comparing evolutionary trees. *Discr. Appl. Math.*, volume 71, pages 153-169.

Hickey, G., F. Dehne, A. Rau-Chaplin, et C. Blouin. (2006). The computational complexity of the unrooted subtree prune and regraft distance. *Technical Report, CS-2006-06*, Dalhousie University.

Ho M.W. (2002) Recent evidence confirms risks of horizontal gene transfer. http://www.i-sis.org.uk/FSAopenmeeting.php

Huson, D.H. (1998). SplitsTree: a program for analyzing and visualizing evolutionary data. *Bioinformatics*, volume 14, pages 68-73.

Huson, D.H. et D. Bryant. (2006). Application of Phylogenetic Networks in Evolutionary Studies. *Mol. Biol. Evol.*, volume 23(2), pages 254-267.

Jain, R., M.C. Rivera, et J. A. Lake. (1999). Horizontal gene transfer among genomes: the complexity hypothesis. *Proc. Natl. Acad. Sci.* U.S.A., volume 96, pages 3801-3806.

Jin, G., L. Nakhleh, S. Snir, et T. Tuller. (2006). Efficient parsimony-based methods for phylogenetic network reconstruction. *Bioinformatics*, volume 23, pages 123-128.

Jin, G., L. Nakhleh, S. Snir, et T. Tuller. (2007). Inferring phylogenetic networks by the maximum parsimony criterion. *Mol. Biol. Evol.*, volume 24:1, pages 324-337.

Jin, L. et M. Nei. (1990). Limitations of the evolutionary parsimony method of phylogenetic analysis. *Mol. Biol. Evol.*, volume 7, pages 82-102.

Jukes, T.H. et C. Cantor. (1969). TTMammalian Protein Metabolism. In *Evolution of protein molecules*, H.N. Munro (Eds), New York: Academic Press, pages 21-132.

Kimura, M.A. (1980). Simple method for estimating evolutionary rate of base substitutions through comparative studies of nucleotide sequences. *J. Mol. Evol.*, volume 16, pages 111-120.

Koonin, E.V. (2003). Horizontal gene transfer: the path to maturity. *Mol. Microbiol.*, volume 50, pages 725-727.

Kuhner, M., et J. Felsenstein. (1994). A simulation comparison of phylogeny algorithms under equal and unequal evolutionary rates. *Mol. Biol. Evol.*, volume 11, pages 459-468.

Greenhill, S.J., T.E. Currie et R.D. Gray. (2009). Does horizontal transmission invalidate cultural phylogenies? *Proc. Royal Soc. London. Series B, Biol. Sc.*, volume 276, pages 2299-2306.

Lake, J.A. et M.C. Rivera. (1999). Horizontal gene transfer among genomes: The complexity hypothesis. *Proceedings of the National Academy of Science*, volume 96(7), pages 3801-380.

Lawrence, J. G., et H. Ochman. (1997). Amelioration of bacterial genomes: rates of change and exchange. *J. Mol. Evol.*, volume 44, pages 383-397.

Legendre, P. (2000). Special section on reticulate evolution. *J. Classif.*, volume 17, pages 153-195.

Legendre, P. et V. Makarenkov. (2002). Reconstruction of biogeographic and evolutionary networks using reticulograms. *Syst. Biol.*, volume 51:2, pages 199-216.

Levenshtein, V.I. (1966). Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady*, volume 10, pages 707-710.
Lerat, E., V. Daubin, et A.N. Moran. (2003). From gene trees to organismal phylogeny in prokaryotes: The case of the γ-proteobacteria. *PLoS. Biology*, volume 1, pages 101-109.

MacLeod, D., R. L. Charlebois, F. Doolittle et E. Bapteste. (2005). Deduction of probable events of lateral gene transfer through comparison of phylogenetic trees by recursive consolidation and rearrangement. *BMC Evol. Biol.*, volume 5:27.

Maddison, D. R., et K. S. Schulz. (2004). The Tree of Life Web Project. Internet address: http://tolweb.org.

Maddison, W. P. (1997). Gene trees in species trees. *Syst. Biol.*, volume 46, pages 523-536.

Maiden, M. (1998). Horizontal genetic exchange, evolution, and spread of antibiotic resistance in bacteria. *Clin. Infect. Dis.*, volume 27, pages 12-20.

Makarenkov, V. (2001). T-Rex: reconstructing and visualizing phylogenetic trees and reticulation networks. *Bioinformatics*, volume 17, pages 664-668.

Makarenkov, V., A. Boc et A.B. Diallo. (2004). Determining horizontal gene transfers in species classification: unique scenario, In *Classification, Clustering, and Data Mining Applications*, IFCS, Springer Verlag, Chicago, pages 439-446.

Makarenkov, V. et P. Legendre. (2004). From a phylogenetic tree to a reticulated network, *J. Comput. Biol.*, volume 11:1, pages 195-212.

Makarenkov, V., A. Boc, C.F. Delwiche, A.B. Diallo, et H. Philippe. (2006). New efficient algorithm for modeling partial and complete gene transfer scenarios. In *Data Science and Classification*, V. Batagelj, H.-H. Bock, A. Ferligoj, et A. Ziberna (Eds.), IFCS, Series: Studies in Classification, Data Analysis, and Knowledge Organization, Springer Verlag, pages 341-349.

Makarenkov, V., A. Boc, et Alpha B. Diallo (2007). La dissimilarité de bipartitions et son utilisation pour détecter les transferts horizontaux de gènes. Actes des 14-emes Rencontres de la Société Francophone de Classification, ENST de Paris, France, pages 90-93.

Makarenkov, V., A. Boc, Alpha B. Diallo et Abdoulaye B. Diallo. (2008). Algorithms for detecting horizontal gene transfers: Theory and practice. In *Data Mining and Mathematical Programming*, P.M. Pardalos et P. Hansen (Eds.), CRM Proceedings and AMS Lecture Notes, volume 45, pages 159-179.

Matte-Tailliez, O., C. Brochier, P. Forterre, et H. Philippe. (2002). Archaeal phylogeny based on ribosomal proteins. *Mol. Biol. Evol.*, volume 19, pages 631-639.

McDade, L.A. (1992). Hybrids and phylogenetic systematics, II. The impact of hybrids on cladistic analysis. *Evolution*, volume 46, pages 1329-1346.

Mirkin, B. G., I. Muchnik, et T.F. Smith. (1995). A Biologically Consistent Model for Comparing Molecular Phylogenies. *J. Comput. Biol.*, volume 2, pages 493-507.

Mirkin, B.G., T.I. Fenner, M.Y. Galperin et E.V. Koonin. (2003). Algorithms for computing parsimonious evolutionary scenarios for genome evolution, the last universal common ancestor and dominance of horizontal gene transfer in the evolution of prokaryotes. *BMC Evol. Biol.*, volume 3:2.

Moret, B.M.E., L. Nakhleh, T. Warnow, C.R. Linder, A. Tholse, A. Padolina, J. Sun et R.E. Timme. (2004). Phylogenetic Networks: Modeling, Reconstructibility and Accuracy. IEEE/ACM Trans. *Comput. Biol. Bioinf.*, volume 1, pages 13-23.

Nakhleh, L., D. Ruths, et L. Wang. (2005). RIATA-HGT: A fast and accurate heuristic for reconstructing horizontal gene transfer. In *Proceedings of the 11th International Computing and Combinatorics Conference*, Kunming, Yunnan, China, pages 84-93

Ogden, C.K. (1930). *Basic English: A General Introduction with Rules and Grammar*. London, Kegan Paul.

Page, R.D.M. (1994). Maps between trees and cladistic analysis of historical associations among genes, organism and areas. *Syst. Biol.*, volume 43, pages 58-77.

Page, R.D.M. et M.A. Charleston. (1998a). From gene to organismal phylogeny: Reconciled trees. *Bioinformatics* , volume 14, pages 819-820

Page, R.D.M. et M.A. Charleston. (1998). Trees within trees: phylogeny and historical associations. *Trends Ecol. Evol.*, volume 13, pages 356-359.

Pagel, M. (2000). In *Time Depth in Historical Linguistics*, C. Renfrew, A. McMahon et L. Trask (Eds.), The McDonald Institute for Archaeological Research, Cambridge, UK., pages 189-207.

Philippe H. (1993). MUST, a computer package of Management Utilities for Sequences and Trees. *Nucl. Acids Res.*, volume 21, pages 5264-5272.

Rexová, K., D. Frynta et J. Zrzavý. (2003). Cladistic analysis of languages: Indo-European classification based on lexicostatistical data. *Cladistics*, volume 19:2, pages 120-127.

Rieseberg, L.H. et J.D. Morefield. (1995). Character expression, phylogenetic reconstruction, and the detection of reticulate evolution. In *Experimental and Molecular Approaches to Plant*

Biosystematics, P.C. Hoch et A.G. Stephenson (Eds.), Monographs in Systematic Botany from the Missouri Botanical Garden, volume 53, pages 333-354.

Rieseberg, L.H. et N.C. Ellstrand. (1993). What can morphological and molecular markers tell us about plant hybridization? Critical Reviews in Plant. *Sciences*, volume 12, pages 213-241.

Robinson, D.R. et L.R. Foulds. (1981). Comparison of phylogenetic trees. *Math Biosci.*, volume 53, pages 131-147.

Saitou, N. et M. Nei. (1987). The neighbor-joining method:a new method for reconstructing phylogenetic trees. *Mol. Biol. Evol.*, volume 4, pages 406–425.

Sawyer S. (1989). Statistical tests for detecting gene conversion. *Mol. Biol. Evol.*, volume 6:5, pages 526-538.

Sneath, P.H.A., M.J. Sackin et R.P. Ambler. (1975). Detecting evolutionary incompatibilities from protein sequences. *Syst. Zool.*, volume 24, pages 311-332.

Sonea, S., et M. Panisset. (1976). Pour une nouvelle bactériologie. *Revue Canadienne de Biologie*, volume 35, pages 103-167.

Stephens, J.C. (1985). Statistical methods of DNA sequence analysis: detection of intragenic recombination or gene conversion. *Mol. Biol. Evol.*, volume 2, pages 539-556.

Strimmer, K., et A. von Haeseler. (1996). Quartet puzzling: a quartet maximum likelihood method for reconstructing tree topologies. *Mol. Biol. Evol.*, volume 13, pages 964-969.

Swadesh, M. (1952). Lexico-statistic dating of prehistoric ethnic contacts. *Proceedings of the American Philosophical Society*, volume 96, pages 452-463.

Templeton, A.R., K.A. Crandall et C.F. Sing. (1992). A cladistic analysis of phenotypic associations with haplotypes inferred from restriction endonuclease mapping and DNA sequence data. III. Cladogram estimation. *Genetics*, volume 132, pages 619-633.

Than, C., D. Ruths, H. Innan, et L. Nakhleh. (2007). Confounding factors in HGT detection: Statistical error, coalescent effects, and multiple solutions. *J. Comp. Biol.*, volume 14, pages 517-535.

Than, C. et L. Nakhleh. (2008). SPR-based tree reconciliation: Non-binary trees and multiple solutions. In *Proceedings of the 6th Asia Pacific Bioinformatics Conference*, Kyoto, Japan. pages 251-260.

Than, C., D. Ruths et L. Nakhleh. (2008). PhyloNet: A software package for analyzing and reconstructing reticulate evolutionary relationships. *BMC Bioinf.*, volume 9:322

Than, C., G. Jin et L. Nakhleh. (2008). Integrating sequence and topology for efficient and accurate detection of horizontal gene transfer. In *Proceedings of the 6th RECOM Comparative Genomics Satellite Workshop*, Paris, France, pages 113-127

Tsirigos, A. et I. Rigoutsos. (2005). A new computational method for the detection of horizontal gene transfer events. *Nucl. Acids Res.*, volume 33, pages 922-933.

The NCBI handbook [Internet]. (2002). Bethesda (MD): National Library of Medicine (US), National Center for Biotechnology Information; 2002 Oct. Chapter 17, The Reference Sequence (RefSeq) Project.

Thompson, J.D., D.G. Higgins et T.J. Gibson. (1994). CLUSTAL W: improving the sensitivity of multiple sequence alignment through sequence weighting, positions-specific gap penalties and weight matrix choice. *Nucl. Acids Res.*, volume 22, pages 4673-4680.

Timmis J. N., Ayliffe M. A., Huang C. Y. et Martin W. (2004). Endosymbiotic gene transfer: organelle genomes forge eukaryotic chromosomes. *Nat. Rev. Genet.* , volume 5, pages 123-135

von Haeseler, A., et G.A. Churchill. (1993). Network models for sequence evolution. *J. Mol. Evol.*, volume 37, pages 77-85.

Webster's Third New International Dictionary, *Unabridged* (Merriam-Webster Inc, new edition, 1960).

Woese, C.R., G. Olsen, M. Ibba, et D. Söll. (2000). Aminoacyl-tRNA synthetases, the genetic code, and the evolutionary process. *Microbiol. Mol. Biol. Rev.*, volume 64, pages 202-236.

Xie, G., C.A. Bonner, J. Song, N.O. Keyhani et R.A. Jensen. (2004). Inter-genomic displacement via lateral gene transfer of bacterial trp operons in an overall context of vertical genealogy. *BMC Biol.*, volume 2:15.

Zhaxybayeva, O., P. Lapierre, et J.P. Gogarten. (2004). Genome mosaicism and organismal lineages. *Trends Genet.*, volume 20, pages 254-260.